

Effective on 12/08/2004.
Fees pursuant to the Consolidated Appropriations Act, 2005 (H.R. 4818).

FEE TRANSMITTAL For FY 2005

Applicant claims small entity status. See 37 CFR 1.27

TOTAL AMOUNT OF PAYMENT (\$ 130.00)

| <i>Complete if Known</i> | |
|--------------------------|--------------------|
| Application Number | 10/696,663 |
| Filing Date | October 28, 2003 |
| First Named Inventor | Uchiumi, Katsuhiro |
| Examiner Name | Unassigned |
| Art Unit | 2186 |
| Attorney Docket No. | 16869K-099200US |

METHOD OF PAYMENT (check all that apply)

Check Credit Card Money Order None Other (please identify): _____
 Deposit Account Deposit Account Number: 20-1430 Deposit Account Name: Townsend and Townsend and Crew LLP

For the above-identified deposit account, the Director is hereby authorized to: (check all that apply)

Charge fee(s) indicated below Charge fee(s) indicated below, except for the filing fee

Charge any additional fee(s) or underpayments of fee(s) under 37 CFR 1.16 and 1.17 Credit any overpayments

WARNING: Information on this form may become public. Credit card information should not be included on this form. Provide credit card information and authorization on PTO-2038

FEE CALCULATION

1. BASIC FILING, SEARCH, AND EXAMINATION FEES

| <u>Application Type</u> | <u>FILING FEES</u> | | <u>SEARCH FEES</u> | | <u>EXAMINATION FEES</u> | | <u>Fees Paid (\$)</u> |
|-------------------------|---------------------|-----------------|---------------------|-----------------|-------------------------|-----------------|-----------------------|
| | <u>Small Entity</u> | <u>Fee (\$)</u> | <u>Small Entity</u> | <u>Fee (\$)</u> | <u>Small Entity</u> | <u>Fee (\$)</u> | |
| Utility | 300 | 150 | 500 | 250 | 200 | 100 | |
| Design | 200 | 100 | 100 | 50 | 130 | 65 | |
| Plant | 200 | 100 | 300 | 150 | 160 | 80 | |
| Reissue | 300 | 150 | 500 | 250 | 600 | 300 | |
| Provisional | 200 | 100 | 0 | 0 | 0 | 0 | |

2. EXCESS CLAIM FEES

Fee Description

Each claim over 20 or, for Reissues, each claim over 20 and more than in the original patent
 Each independent claim over 3 or, for Reissues, each independent claim more than in the original patent
 Multiple dependent claims

| <u>Total Claims</u> | <u>Extra Claims</u> | <u>Fee (\$)</u> | <u>Fee Paid (\$)</u> | <u>Multiple Dependent Claims</u> | |
|--|---------------------|-----------------|----------------------|----------------------------------|----------------------|
| | | | | <u>Fee (\$)</u> | <u>Fee Paid (\$)</u> |
| -20 or HP = | x | = | | | |
| HP = highest number of total claims paid for, if greater than 20 | | | | | |

| <u>Indep. Claims</u> | <u>Extra Claims</u> | <u>Fee (\$)</u> | <u>Fee Paid (\$)</u> |
|----------------------|---------------------|-----------------|----------------------|
| -3 or HP = | x | = | |

HP = highest number of independent claims paid for, if greater than 3

3. APPLICATION SIZE FEE

If the specification and drawings exceed 100 sheets of paper, the application size fee due is \$250 (\$125 for small entity) for each additional 50 sheets or fraction thereof. See 35 U.S.C. 41(a)(1)(G) and 37 CFR 1.16(s).

| <u>Total Sheets</u> | <u>Extra Sheets</u> | <u>Number of each additional 50 or fraction thereof</u> | <u>Fee (\$)</u> | <u>Fee Paid (\$)</u> |
|---------------------|---------------------|---|-----------------|----------------------|
| - 100 = | / 50 = | (round up to a whole number) x | = | |

4. OTHER FEE(S)

Non-English Specification, \$130 fee (no small entity discount)

Other: Petitions to the Commissioner 130.00

SUBMITTED BY

| | | | |
|-------------------|---|---|------------------------|
| Signature |  | Registration No. (Attorney/Agent) 41,405 | Telephone 650-326-2400 |
| Name (Print/Type) | Chun-Pok Leung | | Date January 28, 2005 |



PATENT
Attorney Docket No.: 16869K-099200US
Client Ref. No.: 698 SM/at

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re application of:

KATSUHIRO UCHIUMI et al.

Application No.: 10/696,663

Filed: October 28, 2003

For: CONTROL APPARATUS OF
STORAGE UNIT, AND
METHOD OF CONTROLLING
THE CONTROL APPARATUS
OF STORAGE UNIT

Customer No.: 20350

Examiner: Unassigned

Technology Center/Art Unit: 2186

Confirmation No.: 6841

**PETITION TO MAKE SPECIAL FOR
NEW APPLICATION UNDER M.P.E.P.
§ 708.02, VIII & 37 C.F.R. § 1.102(d)**

Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Sir:

This is a petition to make special the above-identified application under MPEP § 708.02, VIII & 37 C.F.R. § 1.102(d). The application has not received any examination by an Examiner.

(a) The Commissioner is authorized to charge the petition fee of \$130 under 37 C.F.R. § 1.17(i) and any other fees associated with this paper to Deposit Account 20-1430.

02/04/2005 MAHMED1 00000012 201430 10696663

01 FC:1464 130.00 DA

(b) All the claims are believed to be directed to a single invention. If the Office determines that all the claims presented are not obviously directed to a single invention, then Applicants will make an election without traverse as a prerequisite to the grant of special status.

(c) Pre-examination searches were made of U.S. issued patents, including a classification search and a foreign patent database search. The searches were performed on or around December 14, 2004, and were conducted by a professional search firm, Mattingly, Stanger & Malur, P.C. An initial online search was conducted using the EAST database and image retrieval system. The classification search covered Class 710 (subclasses 37 and 38), Class 711 (subclasses 147, 113, and 162), and Class 717 (subclass 168). Because of the large size of these subclasses, keywords were used to narrow of number of documents returned. The foreign patent database search was conducted using Espacenet and Japanese database. The inventors further provided four references considered most closely related to the subject matter of the present application (see reference #6), which was cited in the Information Disclosure Statement filed on October 28, 2003.

(d) The following references, copies of which are attached herewith, are deemed most closely related to the subject matter encompassed by the claims:

- (1) U.S. Patent No. 4,747,047;
- (2) U.S. Patent No. 6,289,398;
- (3) U.S. Patent No. 6,643,750 B2;
- (4) U.S. Patent Publication No. 2002/0229764 A1;
- (5) U.S. Patent Publication No. 2004/0123028 A1; and
- (6) Japanese Patent Publication No. 07-306844.

(e) Set forth below is a detailed discussion of references which points out with particularity how the claimed subject matter is distinguishable over the references.

A. Claimed Embodiments of the Present Invention

The claimed embodiments relate to controlling a storage unit for storing and executing commands.

Independent claim 1 recites a control apparatus of a storage unit. The control apparatus has a first communication port for conducting communication with a computer, a first processor that controls the first communication port, a first storage device that stores a first queue for storing a command for conducting the communication sent from the computer to the first communication port, a first memory that the first processor accesses, a second communication port for conducting communication with the computer, a second processor that controls the second communication port, and a second storage device that stores a second queue for storing a command for conducting communication sent from the computer to the second communication port. The first processor executes the command stored in the first queue to thereby control the communication with the computer. The second processor executes the command stored in the second queue to thereby control the communication with the computer. The control apparatus further comprises a unit causing the second processor to implement execution of the command stored in the first queue; and a unit changing data stored in the first memory while the second processor is being caused to implement execution of the command stored in the first queue.

Independent claim 5 recites a method of controlling a control apparatus of a storage unit, the communication port computer, a first communication port, a first queue for control apparatus having a first for conducting communication with a processor that controls the first a first storage device that stores storing a command for conducting communication port, a first nonvolatile memory that the first processor accesses, a second communication port for conducting communication with the computer, a second processor that controls the second communication port, and a second storage device that stores a second queue for storing a command for conducting communication sent from the computer to the second communication port. The first processor executes the command stored in the first queue to thereby control the communication with the computer. The second processor executes the command stored in the second queue to thereby control the communication with the computer. The method comprises causing the second processor to implement execution of the command stored in the first queue; and changing data stored in the first nonvolatile

memory while the second processor is being caused to implement the execution of the command stored in the first queue.

One of the benefits that may be derived is that changing a microprogram of a disk control apparatus can be performed without stopping data transfer between the computer and the disk control apparatus and without the need for changes to the data transfer bus between the computer and the disk control apparatus.

B. Discussion of the References

1. U.S. Patent No. 4,747,047

This reference discloses a data transfer system using two peripheral controllers to access dual-ported data storage units. A data transfer network is provided whereby each of a multiplicity of disk drive units have dual port elements to permit several additional desirable functions to be accomplished, namely to permit faster access to a disk unit by a "second" data link processor (I/O controller) once a "first" data link processor has completed its access to a particular disk drive unit, and to gain access to a particular disk drive unit by a "second" data link processor when a "first" data link processor is inoperative or has lost its capability of accessing the particular disk drive unit. The network involves disk drive units having dual ports which enable at least two data link processors to access any given disk drive unit. In the system, the host adapter unit receives the UO command descriptors from the host system, verifies their correctness and checks their parity, and provides a queue list in a buffer memory of the host adapter, which indicates the jobs or tasks to be accomplished with each of the eight disk drive units. See, e.g., Abstract; Figures 1-8; column 3, lines 5-60; column 4, lines 29-50 and 65-67; and column 6, lines 29-68.

The reference, however, does not teach the use of a unit causing a second processor to implement execution of a command stored in a first queue and a unit changing data stored in a first memory while the second processor is being caused to implement execution of the command stored in the first queue, as recited in independent claims 1 and 5.

2. U.S. Patent No. 6,289,398

This reference relates to a distributed storage array system having a plurality of storage devices, wherein each of the devices include a modular control unit for exchanging configuration information over a communication link. The system includes a plurality of modular data storage devices for storing and retrieving data in a selected data sequence; a communications link; and a plurality of modular control units, each configured to communicate with the host computer directly over the host bus. A first modular control unit and a second modular control unit are programmed to exchange information over the communications link for performance of read/write access of a host computer to a RAID set including a transfer of control in sequence from the first modular control unit to the second modular control unit. See, e.g., Abstract; Figures 1-8; column 4, lines 40-67, column 5; and column 8, line 41 to column 9, line 64.

The reference, however, does not teach the use of a unit causing the second processor to implement execution of the command stored in the first queue, with a unit changing data stored in the first memory while the second processor is being caused to implement execution of the command stored in the first queue, as recited in independent claims 1 and 5.

3. U.S. Patent No. 6,643,750 B2

This reference discloses a storage apparatus system and method of data backup. The storage system includes a first storage apparatus that has a first disk controller for receiving data from a host computer and one or more first disks, each of which is in data communication with the first disk controller. A second storage apparatus is included that has a second disk controller and one or more second disks, each of which is coupled to the second disk controller. A network has the first storage apparatus and the second storage apparatus coupled to it, and the first disk controller is configured to store data received from the host computer into a first storage area of the first storage apparatus, the first disk controller is further configured to send data stored in the first storage area to the second disk controller, while the second disk controller is configured to store data received from the first disk controller into a third storage area in the second storage apparatus. See, e.g., Abstract; Figures 1-28; and column 3, line 32 to column 4, line 12.

The reference, however, does not teach the use of a storage device that stores a first queue for storing a command for conducting the communication sent from a computer to a communication port, as recited in independent claims 1 and 5.

4. U.S. Patent Publication No. 2002/0229764 A1

This reference discloses a data storage subsystem wherein the storage subsystem is coupled to a host computer, and includes a first communication interface coupled to the first host computer via a first communication link. A storage controller is included for handling data requests from the first host computer. A storage unit is coupled to the storage controller and includes a storage area. A second communication interface is coupled to a remote storage subsystem via a second communication link that is different from the first communication link. The storage subsystem is configured to transmit a first data block from the storage subsystem to the remote storage subsystem and to transmit attribute information from the storage subsystem to the remote storage subsystem without intervention from the first host computer. The remote storage subsystem is associated with a second host computer. See, e.g., Abstract; Figures 1-9; and paragraphs [0024]-[0027], [0039]-[0043], [0063]-[0064], and [0088].

The reference, however, does not teach the use of a storage device that stores a first queue for storing a command for conducting the communication sent from a computer to a communication port, as recited in independent claims 1 and 5.

5. U.S. Patent Publication No. 2004/0123028 A1

This reference relates to storage control apparatus, storage system, control method of storage control apparatus, channel control unit and program. The system provides a storage control apparatus including a plurality of channel control units each having an interface with an information processor. The storage control apparatus includes a disk control unit having an interface with a storage device for storing data, a cache memory for temporarily storing data to be interchanged between the information processor and the storage device, and an internal connector unit connecting the channel control units and the disk control unit to each other. Furthermore, the cache memory is disposed in the channel control units that are connected to one another through a dedicated data transfer path used for

storing mutually the data stored in the cache memory. See, e.g., Abstract; Figures 1-25; and paragraphs [0012]-[0013], [0042]-[0055], and [0081]-[0095].

The reference, however, does not teach the use of a storage device that stores a first queue for storing a command for conducting the communication sent from a computer to a communication port, as recited in independent claims 1 and 5.

6. Japanese Patent Publication No. 07-306844

The reference discloses a technique to modify program that a processor group executes without stopping the processor group entirely. The processor group is grouped into two clusters: (1 and 2) and a data transfer control job is normally decentralized to and processed by all the processors irrelevantly to the clusters, when a data transfer control program is modified, a mode wherein the job is decentralized to and processed by only the processors in the same cluster is entered first and when the mode transition is successful, the processing of the cluster 1 is stopped while the processing of the cluster 2 is carried on. Data transfer control programs in respective local memories LM of the processors in the cluster 1 are modified and after the modification, the processing is restarted and then the processing of the cluster 2 is stopped while the processing of the cluster 1 is carried on to modify data transfer control programs in the respective local memories of the processes in the cluster 2 and after the modification, the processing is restarted. Then normal operation wherein the job is decentralized to and processed by all the processors irrelevantly to the clusters is started.

The reference, however, does not teach the use of a unit causing a second processor to implement execution of a command stored in a first queue and a unit changing data stored in a first memory while the second processor is being caused to implement execution of the command stored in the first queue, as recited in independent claims 1 and 5.

(f) In view of this petition, the Examiner is respectfully requested to issue a first Office Action at an early date.

Respectfully submitted,



Chun-Pok Leung
Reg. No. 41,405

TOWNSEND and TOWNSEND and CREW LLP
Two Embarcadero Center, 8th Floor
San Francisco, California 94111-3834
Tel: 650-326-2400
Fax: 415-576-0300
Attachments
RL:rl
60398869 v1

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 07-306844
 (43)Date of publication of application : 21.11.1995

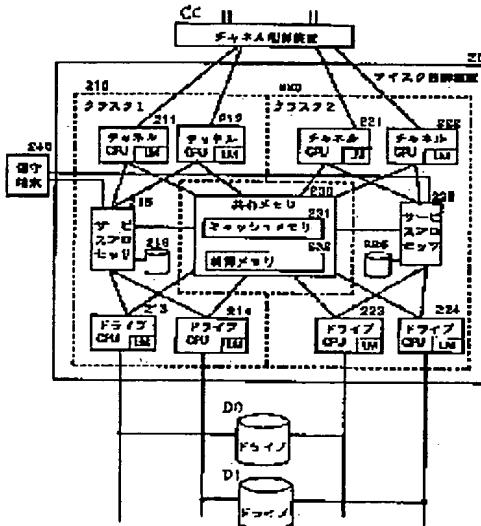
(51)Int.Cl. G06F 15/177
 G06F 9/06

(21)Application number : 06-099705 (71)Applicant : HITACHI LTD
 (22)Date of filing : 13.05.1994 (72)Inventor : MURATA TOMOHIRO
 NITSUTA MITSUAKI
 KURIHARA KENZO

(54) MULTIPROCESSOR SYSTEM AND ITS NONSTOP PROGRAM MODIFYING METHOD, AND DISK CONTROLLER AND ITS NONSTOP PROGRAM MODIFYING METHOD

(57)Abstract:

PURPOSE: To modify programs that a processor group executes without stopping the processor group entirely.
 CONSTITUTION: The processor group is grouped into two clusters (1 and 2) and a data transfer control job is normally decentralized to and processed by all the processors irrelevantly to the clusters, when a data transfer control program is modified, a mode wherein the job is decentralized to and processed by only the processors in the same cluster is entered first and when the mode transition is successful, the processing of the cluster 1 is stopped while the processing of the cluster 2 is carried on. Data transfer control programs in respective local memories LM of the processors in the cluster 1 are modified and after the modification, the processing is restarted and then the processing of the cluster 2 is stopped while the processing of the cluster 1 is carried on to modify data transfer control programs in the respective local memories of the processes in the cluster 2 and after the modification, the processing is restarted. Then normal operation wherein the job is decentralized to and processed by all the processors irrelevantly to the clusters is started.



LEGAL STATUS

[Date of request for examination] 25.08.1998
 [Date of sending the examiner's decision of rejection]
 [Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]
 [Date of final disposal for application]
 [Patent number] 3160149
 [Date of registration] 16.02.2001
 [Number of appeal against examiner's decision of rejection]
 [Date of requesting appeal against examiner's decision of rejection]
 [Date of extinction of right]

* NOTICES *

Japan Patent Office is not responsible for any damages caused by the use of this translation.

1.This document has been translated by computer. So the translation may not reflect the original precisely.

2.**** shows the word which can not be translated.

3.In the drawings, any words are not translated.

CLAIMS

[Claim(s)]

[Claim 1] In the multiprocessor system which performs the program which stored in each local memory of two or more processors, respectively, and each processor stored in self-local memory, respectively, and processes a job. It is the cluster N from a cluster 1 (N) about two or more above-mentioned processors. A group division is carried out or more by two at N clusters to the natural number below [all] the number of processors. And the cluster limited mode in which distributed processing of a job is performed only between the processor groups belonging to the same cluster as the cluster to which a self-processor belongs, and cluster the mode in which it does not limit in which distributed processing of a job is performed among all processors are formed. At the time of normal operation, distributed processing of the job is carried out in cluster the mode in which it does not limit, at the time of program change completed. The program stored in each local memory of the processor group of a selection cluster if the job processed by the processor group of a selection cluster is lost is changed into a new program, respectively. If change of the program of all the processors of a selection cluster is completed, registration of the new job in the processor group of a selection cluster and processing of a job will be resumed. The non-stopped program change method of the multiprocessor system characterized by returning processing of the job in cluster limited mode to processing of the job in cluster the mode in which it does not limit, and returning to normal operation after repeating this in order about all clusters.

[Claim 2] In the multiprocessor system which performs the program which stored in each local memory of two or more processors, respectively, and each processor stored in self-local memory, respectively, and processes a job. It is the cluster N from a cluster 1 (N) about two or more above-mentioned processors. A group division is carried out or more by two at N clusters to the natural number below [all] the number of processors. And the cluster limited mode in which distributed processing of a job is performed only between the processor groups belonging to the same cluster as the cluster to which the group division of the cluster group is carried out at two sets, the 1st set and the 2nd set, and a self-processor belongs. Cluster the mode in which it does not limit in which distributed processing of a job is performed between the processor groups belonging to the cluster which belongs to the same group as the group to which the cluster to which the processor group and self-processor belonging to the same cluster as the cluster to which a self-processor belongs belong, and ****. At a distributed-processing modal-control operation All clusters are made to belong to either the 1st set or the 2nd set, and distributed processing of the job is carried out in cluster the mode in which it does not limit, at the time of program change operation First, processing of the job in cluster the mode in which it does not limit in a selection cluster is shifted to processing of the job in one cluster and continuing distributed processing of the job in cluster the mode in which it does not limit in a non-choosing cluster. If shift is completed, registration of the new job in the

processor group of a selection cluster will be stopped. The program stored in each local memory of the processor group of a selection cluster if the job processed by the processor group of a selection cluster is lost is changed into a new program, respectively. If change of the program of all the processors of a selection cluster is completed, while resuming registration of the new job in the processor group of a selection cluster, and processing of a job After changing affiliation of a selection cluster into another side from either the 1st set or the 2nd set, processing of the job in the cluster limited mode in a selection cluster is returned to processing of the job in cluster the mode in which it does not limit. The non-stopped program change method of the multiprocessor system characterized by repeating this in order about all clusters, and returning to normal operation.

[Claim 3] In the multiprocessor system which performs the program which stored in each local memory of two or more processors, respectively, and each processor stored in self-local memory, respectively, and processes a job. It is the cluster N from a cluster 1 (N) about two or more above-mentioned processors. The cluster quota means which carries out a group division or more by two at N clusters to the natural number below [all] the number of processors. So that, distributed processing may be performed by the cluster limited mode in which distributed processing of a job is performed only between the processor groups belonging to the same cluster as the cluster to which a self-processor belongs, cluster the mode in which it does not limit in which distributed processing of a job is performed among all processors, and *****. At a distributed-processing modal-control means to control, and the time of normal operation The aforementioned cluster mode in which it does not limit is made to perform distributed processing of a job. at the time of program change operation First, it is made to shift to processing of the job in cluster limited mode from processing of the job in cluster the mode in which it does not limit. Registration of the new job in the processor group of a selection cluster is stopped choosing one cluster and making processing of the job in the processor group of a non-choosing cluster, and registration of a new job continue, when shift is completed. The program stored in each local memory of the processor group of a selection cluster if the job processed by the processor group of a selection cluster is lost is changed into a new program, respectively. When completing change of the program of all the processors of a selection cluster, registration of the new job in the processor group of a selection cluster and processing of a job are made to resume. The multiprocessor system characterized by providing a program change processing means to make processing of the job in cluster limited mode shift to processing of the job in cluster the mode in which it does not limit, and to make normal operation return after repeating this in order about all clusters.

[Claim 4] In the multiprocessor system which performs the program which stored in each local memory of two or more processors, respectively, and each processor stored in self-local memory, respectively, and processes a job. It is the cluster N from a cluster 1 (N) about two or more above-mentioned processors. The cluster quota means which carries out a group division or more by two at N clusters to the natural number below [all] the number of processors. The distributed-processing possible cluster combination means which carries out the group division of the cluster group at two sets, the 1st set and the 2nd set. In the cluster limited mode in which distributed processing of a job is performed only between the processor groups belonging to the same cluster as the cluster to which a self-processor belongs So that distributed processing may be performed by cluster the mode in which it does not limit in which distributed processing of a job is performed between the processor groups belonging to the cluster which belongs to the same group as the group to which the cluster to which the processor group and self-processor belonging to the same cluster as the cluster to which a self-processor belongs belong, and ****. At a distributed-processing modal-control means to control, and the time of normal operation All clusters are made to belong to either the 1st set or the 2nd set, and distributed processing of the job is carried out in cluster the mode in which it does not limit, at the time of program change operation First, processing of the job in cluster the mode in which it does not limit in a selection cluster is shifted to processing of the job in cluster limited mode, choosing one cluster and making distributed processing of the job in cluster the mode in which it does not limit in a non-choosing cluster continue. If shift is

completed, registration of the new job in the processor group of a selection cluster will be stopped. The program stored in each local memory of the processor group of a selection cluster if the job processed by the processor group of a selection cluster is lost is changed into a new program, respectively. When completing change of the program of all the processors of a selection cluster, while making registration of the new job in the processor group of a selection cluster, and processing of a job resume Processing of the job in the cluster limited mode in a selection cluster is made to shift to processing of the job in cluster the mode in which it does not limit, after making affiliation of a selection cluster changed into another side from either the 1st set or the 2nd set. The multiprocessor system characterized by providing a program change processing means to repeat this in order about all clusters, and to make normal operation return.

[Claim 5] The multiprocessor system which performs the program which stored the program in each local memory of two or more processors characterized by providing the following, respectively, and each processor stored in self-local memory, respectively, and processes a job A cluster allocation information-storage means to memorize the cluster allocation information which shows whether each processor belongs to which cluster in order to carry out the group division of two or more above-mentioned processors at two or more clusters A mode-of-operation directions information-storage means to memorize the cluster mode-of-operation directions information which shows per cluster whether it is made the normal mode of operation to which a processor processes a job according to the program stored in local memory, the pause mode in which a processor stops processing of a job since a cluster is blockaded, the maintenance mode of operation which changes into a new program the program stored in local memory, and the mode of operation of ***** An execution cluster directions information-storage means to memorize the execution cluster directions information which shows whether a job is processed in the cluster limited mode in which distributed processing of the processor belonging to the same cluster as the cluster to which a self-processor belongs, and a job is performed, cluster the mode in which it does not limit, in which adhere to a cluster and distributed processing of a job is performed among arbitrary processors [be / nothing] and the distributed-processing mode of ***** When the change start of a program is instructed to be a program change start directions means to direct the change start of a program. Set distributed-processing mode as cluster limited mode, and one arbitrary cluster is chosen. Set the mode of operation of the cluster as pause mode, and when all the processors belonging to the cluster concerned will be in hibernation, the mode of operation of the cluster concerned is set as a maintenance mode of operation. The program stored in the local memory of all the processors belonging to the cluster concerned is changed into a new program. A program change processing means to repeat this about all clusters, to be at the completion time, to set the mode of operation of all clusters as a normal mode of operation, and to set distributed-processing mode as cluster the mode in which it does not limit

[Claim 6] The multiprocessor system which performs the program which stored the program in each local memory of two or more processors characterized by providing the following, respectively, and each processor stored in self-local memory, respectively, and processes a job A cluster allocation information-storage means to memorize the cluster allocation information which shows whether each processor belongs to which cluster in order to carry out the group division of two or more above-mentioned processors at two or more clusters A mode-of-operation directions information-storage means to memorize the cluster mode-of-operation directions information which shows per cluster whether it is made the normal mode of operation to which a processor processes a job according to the program stored in local memory, the pause mode in which a processor stops processing of a job since a cluster is blockaded, the maintenance mode of operation which changes into a new program the program stored in local memory, and the mode of operation of ***** An execution cluster directions information-storage means to memorize the execution cluster directions information which shows whether a job is processed in cluster the mode in which it does not limit, in which distributed processing of a job is possible between the processors and self-processors belonging to the same cluster as the processor belonging to the same cluster as the cluster to which a self-processor belongs, and

the cluster to which the cluster limited mode in which distributed processing of a job is performed, and a self-processor belongs, and also distributed processing of a job is performed by the processor of a cluster, and the When the change start of a program is instructed to be a program change start directions means to direct the changes start of a program. Choose one arbitrary cluster and the distributed-processing mode of the cluster is set as cluster limited mode. Set the mode of operation of the cluster concerned as pause mode, and when all the processors belonging to the cluster concerned will be in hibernation, the mode of operation of the cluster concerned is set as a maintenance mode of operation. The program stored in the local memory of all the processors belonging to the cluster concerned is changed into a new program. A program change processing means to set the distributed-processing mode of the cluster concerned as cluster the mode in which it does not limit, to repeat this about all clusters, to be at the completion time, and to set the mode of operation of all clusters as a normal mode of operation

[Claim 7] A data transfer control program is stored in each local memory of two or more processors, respectively. In the disk controller which performs the data transfer control program which each processor stored in self-local memory, respectively, and processes the data transfer job over a disk drive It is the cluster N from a cluster 1 (N) about two or more above-mentioned processors. A group division is carried out or more by two at N clusters to the natural number below [all] the number of processors. The cluster limited mode in which distributed processing of a data transfer job is performed only between the processor groups belonging to the same cluster as the cluster to which a self-processor belongs, and cluster the mode in which it does not limit in, which distributed processing of a data transfer job is performed among all processors are formed, and at the time of normal operation Distributed processing of the data transfer job is carried out in cluster the mode in which it does not limit, at the time of data transfer control program change operation First, it shifts to processing of the data transfer job in cluster limited mode from processing of the data transfer job in cluster the mode in which it does not limit. Registration of the new data transfer job in the processor group of a selection cluster is stopped choosing one cluster and continuing processing of the data transfer job in the processor group of a non-choosing cluster, and registration of a new data transfer job, if shift, is completed. The data transfer control program stored in each local memory of the processor group of a selection cluster if the data transfer job processed by the processor group of a selection cluster is lost is changed into a new data transfer control program, respectively. If change of the data transfer control program of all the processors of a selection cluster is completed, registration of the new data transfer job in the processor group of a selection cluster and processing of a data transfer job will be resumed. The non-stopped program change method of the disk controller characterized by returning processing of the data transfer job in cluster limited mode to processing of the data transfer job in cluster the mode in which it does not limit, and returning to normal operation after repeating this in order about all clusters.

[Claim 8] A data transfer control program is stored in each local memory of two or more processors, respectively. In the disk controller with which each processor performs the data transfer control program stored in self-local memory, respectively, and processes a data transfer job It is the cluster N from a cluster 1 (N) about two or more above-mentioned processors. A group division is carried out or more by two at N clusters to the natural number below [all] the number of processors. And the cluster limited mode in which distributed processing of a data transfer job is performed only between the processor groups belonging to the same cluster as the cluster to which the group division of the cluster group is carried out at two sets, the 1st set and the 2nd set, and a self-processor belongs. Cluster the mode in which it does not limit in, which distributed processing of a data transfer job is performed between the processor groups belonging to the cluster which belongs to the same group as the group to which the cluster to which the processor group and self-processor belonging to the same cluster as the cluster to which a self-processor belongs is formed, at the time of normal operation All clusters are made to belong to either the 1st set or the 2nd set, and distributed processing of the data transfer job is carried out in cluster the mode in which it does not limit, at the time of data transfer control program change operation First, processing of the data transfer job in

cluster the mode in which it does not limit in a selection cluster is shifted to processing of the data transfer job in cluster limited mode, choosing one cluster and continuing distributed processing of the data transfer job in cluster the mode in which it does not limit in a non-choosing cluster. If shift is completed, registration of the new data transfer job in the processor group of a selection cluster will be stopped. The data transfer control program stored in each local memory of the processor group of a selection cluster if the data transfer job processed by the processor group of a selection cluster is lost is changed into a new data transfer control program, respectively. If change of the data transfer control program of all the processors of a selection cluster is completed, while resuming registration of the new data transfer job in the processor group of a selection cluster, and processing of a data transfer job After changing affiliation of a selection cluster into another side from either the 1st set or the 2nd set, processing of the data transfer job in the cluster limited mode in a selection cluster is returned to processing of the data transfer job in cluster the mode in which it does not limit. The non-stopped program change method of the disk controller characterized by repeating this in order about all clusters, and returning to normal operation.

[Claim 9] A data transfer control program is stored in each local memory of two or more processors, respectively. In the disk controller with which each processor performs the data transfer control program stored in self-local memory, respectively, and processes a data transfer job in the cluster N from a cluster 1 (N) about two or more above-mentioned processors. The cluster quota means which carries out a group division or more by two at N clusters to the natural number below [all] the number of processors. So that distributed processing may be performed by the cluster limited mode in which distributed processing of a data transfer job is performed only between the processor groups belonging to the same cluster as the cluster to which a self-processor belongs, cluster the mode in which it does not limit in,which distributed processing of a data transfer job is performed among all processors, and *****. At a distributed-processing modal-control means to control, and the time of normal operation The aforementioned cluster mode in which it does not limit is made to perform distributed processing of a data transfer job. at the time of data transfer control program change operation First, it is made to shift to processing of the data transfer job in cluster limited mode from processing of the data transfer job in cluster the mode in which it does not limit. Registration of the new data transfer job in the processor group of a selection cluster is stopped choosing one cluster and making processing of the data transfer job in the processor group of a non-choosing cluster, and registration of a new data transfer job continue, when shift is completed. The data transfer control program stored in each local memory of the processor group of a selection cluster if the data transfer job processed by the processor group of a selection cluster is lost is changed into a new data transfer control program, respectively. When completing change of the data transfer control program of all the processors of a selection cluster, registration of the new data transfer job in the processor group of a selection cluster and processing of a data transfer job are made to resume. After repeating this in order about all clusters, The disk controller characterized by providing a data transfer control program change processing means to make processing of the data transfer job in cluster limited mode shift to processing of the data transfer job in cluster the mode in which it does not limit, and to make normal operation return.

[Claim 10] The disk controller which performs the data transfer control program which is characterized by providing the following, and which stored the data transfer control program in each local memory of two or more processors, respectively, and each processor stored in self-local memory, respectively, and processes a data transfer job The cluster quota means which carries out the group division of two or more above-mentioned processors at N clusters from a cluster 1 to Cluster N (N is the natural number below [all] the number of processors or more in two) The distributed-processing possible cluster combination means which carries out the group division of the cluster group at two sets, the 1st set and the 2nd set A means [modal control / distributed processing controlled to perform distributed processing by cluster the mode in which it does not limit, in which distributed processing of a data transfer job is performed between the processor groups belonging to the cluster which belongs to the same group as the group to which the cluster to which the processor group and self-processor belonging to the same cluster

as the cluster to which the cluster limited mode in which distributed processing of a data transfer job is performed only between the processor groups belonging to the same cluster as the cluster to which a self-processor belongs, and a self-processor belongs to either the 1st set or the 2nd set, and carries out normal operation makes all clusters belong to either the 1st set or the 2nd set, and carries out distributed processing of the data transfer job in cluster the mode in which it does not limit, at the time of data transfer control program change operation First, processing of the data transfer job in cluster the mode in which it does not limit in a selection cluster is made to shift to processing of the data transfer job in cluster limited mode, choosing one cluster and making distributed processing of the data transfer job in cluster the mode in which it does not limit in a non-choosing cluster continue. If shift is completed, registration of the new data transfer job in the processor group of a selection cluster will be stopped. The data transfer control program stored in each local memory of the processor group of a selection cluster if the data transfer job processed by the processor group of a selection cluster is lost is changed into a new data transfer control program, respectively. When completing change of the data transfer control program of all the processors of a selection cluster, while making registration of the new data transfer job in the processor group of a selection cluster, and processing of a data transfer job resume Processing of the data transfer job in the cluster limited mode in a selection cluster is made to shift to processing of the data transfer job in cluster the mode in which it does not limit, after making affiliation of a selection cluster change into another side from either the 1st set or the 2nd set. A data transfer control program change processing means to repeat this in order about all clusters, and to make normal operation return

[Claim 11] The disk controller which performs the data transfer control program which is characterized by providing the following, and which stored the data transfer control program in each local memory of two or more processors, respectively, and each processor stored in self-local memory, respectively, and processes a data transfer job A cluster allocation information-storage means to memorize the cluster allocation information which shows whether each processor belongs to which cluster in order to carry out the group division of two or more above-mentioned processors at two or more clusters A mode-of-operation directions information-storage means to memorize the cluster mode-of-operation directions information which shows's per cluster whether the normal mode of operation to which a processor processes a data transfer job according to the data transfer control program stored in local memory, the pause mode in which a processor stops processing of a data transfer job since a cluster is blockaded, and the data transfer control program stored in local memory is made into the maintenance mode of operation changed into a new data transfer control program, and the mode of operation of ***** An execution cluster directions information-storage means to memorize the execution cluster directions information which shows whether a data transfer job is processed in the cluster limited mode in which distributed processing of the processor belonging to the same cluster as the cluster to which a self-processor belongs, and a data transfer job is performed, cluster the mode in which it does not limit in which adhere to a cluster and distributed processing of a data transfer job is performed among arbitrary processors [be / nothing], and the distributed-processing mode of ***** When the change start of a data transfer control program is instructed to be a data transfer control program change start, directions means to direct the change start of a data transfer control program. Set distributed-processing mode as cluster limited mode, and one arbitrary cluster is chosen. Set the mode of operation of the cluster as pause mode, and when all the processors belonging to the cluster concerned will be in hibernation, the mode of operation of the cluster concerned is set as a maintenance mode of operation. The data transfer control program stored in the local memory of all the processors belonging to the cluster concerned is changed into a new data transfer control program. A data transfer control program change processing means to repeat this about all clusters, to be at the completion time, to set the mode of operation of all clusters as a normal mode of operation, and to set distributed-processing mode as cluster the mode in which it does not limit

[Claim 12] The disk controller which performs the data transfer control program which is characterized by providing the following, and which stored the data transfer control program in

each local memory of two or more processors, respectively, and each processor stored in self-local memory, respectively, and processes a data transfer job. A cluster allocation information-storage means to memorize the cluster allocation information which shows whether each processor belongs to which cluster in order to carry out the group division of two or more above-mentioned processors at two or more clusters. A mode-of-operation directions information-storage means to memorize the cluster mode-of-operation directions information which shows per cluster whether the normal mode of operation to which a processor processes a data transfer job according to the data transfer control program stored in local memory, the pause mode in which a processor stops processing of a data transfer job since a cluster is blocked, and the data transfer control program stored in local memory is made into the maintenance mode of operation changed into a new data transfer control program, and the mode of operation of *****. An execution cluster directions information-storage means to memorize the execution cluster directions information which shows whether a data-transfer job is processed in cluster the mode in which it does not limit that distributed processing of a data transfer job is possible between the processors and self-processors belonging to the same cluster as the processor belonging to the same cluster as the cluster to which a self-processor belongs, and the cluster to which the cluster limited mode in which distributed processing of a data transfer job is performed, and a self-processor belongs, and also distributed processing of a data-transfer job is performed by the processor of. When the change start of a data transfer control program is instructed to be a data transfer control program change start directions means to direct the change-start of a data transfer control program. Choose one arbitrary cluster and the distributed-processing mode of the cluster is set as cluster limited mode. Set the mode of operation of the cluster concerned as pause mode, and when all the processors belonging to the cluster concerned will be in hibernation, the mode of operation of the cluster concerned is set as a maintenance mode of operation. The data transfer control program stored in the local memory of all the processors belonging to the cluster concerned is changed into a new data transfer control program. A data transfer control program change processing means to set the distributed-processing mode of the cluster concerned as cluster the mode in which it does not limit, to repeat this about all clusters, to be at the completion time, and to set the mode of operation of all clusters as a normal mode of operation

[Translation done.]

* NOTICES *

Japan Patent Office is not responsible for any damage caused by the use of this translation.

- 1.This document has been translated by computer. So the translation may not reflect the original precisely.
- 2.**** shows the word which can not be translated.
- 3.In the drawings, any words are not translated.

DETAILED DESCRIPTION

[Detailed Description of the Invention]

[0001] [Industrial Application] this invention relates to the non-stopped program change method of a multiprocessor system, a multiprocessor system, the non-stopped program change method of a disk controller, and a disk controller. It is related with the non-stopped program change method of a multiprocessor system that the program which a processor group performs can be changed, a multiprocessor system, the non-stopped program change method of a disk controller, and a disk controller, in more detail, without stopping a processor group completely.

[0002] [Description of the Prior Art] The main ** processor and a * system processor are prepared, and after performing the backup system change which succeeds the processing which was being performed by the main ** processor or by the ** system processor, the non-stopped maintenance procedure which is made to stop the main ** processor and is maintained is indicated by JP-56-52460,A.

[0003] [Problem(s) to be Solved by the Invention] According to the above-mentioned conventional non-stopped maintenance procedure, it is possible to maintain the system portion belonging to the main ** processor, without stopping processing completely. However, in order to succeed the processing which was being performed by the main ** processor by the ** system processor, it is required for the program which the main ** processor and a ** system processor perform, respectively to be the same version, and in upgrading a program, there is a trouble which cannot apply the above-mentioned non-stopped maintenance procedure. That is, in a Prior art, in upgrading a program, there is a trouble that a processor group must be stopped completely. Then, the purpose of this invention is to offer the non-stopped program change method of a multiprocessor system that the program which a processor group performs can be changed, a multiprocessor system, the non-stopped program change method of a disk controller, and a disk controller, without stopping a processor group completely.

[0004] [Means for Solving the Problem] In the 1st viewpoint, this invention stores a program in each local memory of two or more processors, respectively. In the multiprocessor system with which each processor performs the program stored in self-local memory, respectively, and processes a job it is the cluster N from a cluster 1 (N) about two or more above-mentioned processors. A group division is carried out or more by two at N clusters to the natural number below [all] the number of processors. And the cluster limited mode in which distributed processing of a job is performed only between the processor groups belonging to the same cluster as the cluster to which a self-processor belongs, and distributed processing of a job is performed among all processors are formed. At the time of normal operation, distributed processing of the job is carried out in cluster the mode in which it does not limit, at the time of program change operation First, it shifts to processing of the job in cluster limited mode from processing of the job in cluster the mode in which it does not limit. Registration of the new job in the processor group of a selection cluster is stopped choosing one

cluster and continuing processing of the job in the processor group of a non-choosing cluster, and registration of a new job, if shift is completed. The program stored in each local memory of the processor group of a selection cluster if the job processed by the processor group of a selection cluster is lost is changed into a new program, respectively. If change of the program of all the processors of a selection cluster is completed, registration of the new job in the processor group of a selection cluster and processing of a job will be resumed. After repeating this in order about all clusters, the non-stopped program change method of the multiprocessor system characterized by returning processing of the job in cluster limited mode to processing of the job in cluster the mode in which it does not limit, and returning to normal operation is offered.

[0005] In the 2nd viewpoint, this invention stores a program in each local memory of two or more processors, respectively. In the multiprocessor system with which each processor performs the program stored in self-local memory, respectively, and processes a job It is the cluster N from a cluster 1 (N) about two or more above-mentioned processors. A group division is carried out or more by two at N clusters to the natural number below [all] the number of processors. And the cluster limited mode in which distributed processing of a job is performed only between the processor groups belonging to the same cluster as the cluster to which the group division of the cluster group is carried out at two sets, the 1st set and the 2nd set, and a self-processor belongs. Cluster the mode in which it does not limit in which distributed processing of a job is performed between the processor groups belonging to the cluster which belongs to the same group as the group to which the cluster to which the processor group and self-processor belonging to the same cluster as the cluster to which a self-processor belongs belong is formed, at the time of normal operation All clusters are made to belong to either the 1st set or the 2nd set, and distributed processing of the job is carried out in cluster the mode in which it does not limit, at the time of program change operation First, processing of the job in cluster the mode in which it does not limit in a selection cluster is shifted to processing of the job in cluster limited mode, choosing one cluster and continuing distributed processing of the job in cluster the mode in which it does not limit in a non-choosing cluster. If shift is completed, registration of the new job in the processor group of a selection cluster will be stopped. The program stored in each local memory of the processor group of a selection cluster if the job processed by the processor group of a selection cluster is lost is changed into a new program, respectively. If change of the program of all the processors of a selection cluster is completed, while resuming registration of the new job in the processor group of a selection cluster, and processing of a job After changing affiliation of a selection cluster into another side from either the 1st set or the 2nd set, processing of the job in the cluster limited mode in a selection cluster is returned to processing of the job in cluster the mode in which it does not limit. The non-stopped program change method of the multiprocessor system characterized by repeating this in order about all clusters, and returning to normal operation is offered.

[0006] In the 3rd viewpoint, this invention offers the multiprocessor system which enforces the non-stopped program change method by the 1st viewpoint of the above, or the non-stopped program change method by the 2nd viewpoint of the above.

[0007] In the 4th viewpoint, this invention stores a data transfer control program in each local memory of two or more processors, respectively. In the disk controller which performs the data transfer control program which each processor stored in self-local memory, respectively, and processes the data transfer job over a disk drive it is the cluster N from a cluster 1 (N) about two or more above-mentioned processors. A group division is carried out or more by two at N clusters to the natural number below [all] the number of processors. The cluster limited mode in which distributed processing of a data transfer job is performed only between the processor groups belonging to the same cluster as the cluster to which a self-processor belongs, and cluster the mode in which it does not limit in which distributed processing of a data transfer job is performed among all processors are formed, and at the time of normal operation Distributed processing of the data transfer job is carried out in cluster the mode in which it does not limit, at the time of data transfer control program change operation First, it shifts to processing of the data transfer job in cluster limited mode from processing of the data transfer job in cluster the

mode in which it does not limit. Registration of the new data transfer job in the processor group of a selection cluster is stopped choosing one cluster and continuing processing of the data transfer job in the processor group of a non-choosing cluster, and registration of a new data transfer job, if shift is completed. The data transfer control program stored in each local memory of the processor group of a selection cluster if the data transfer job processed by the processor group of a selection cluster is lost is changed into a new data transfer control program, respectively. If change of the data transfer control program of all the processors of a selection cluster is completed, registration of the new data transfer job in the processor group of a selection cluster and processing of a data transfer job will be resumed. After repeating this in order about all clusters, the non-stopped program change method of the disk controller characterized by returning processing of the data transfer job in cluster limited mode to processing of the data transfer job in cluster the mode in which it does not limit, and returning to normal operation is offered.

[0008] In the 5th viewpoint, this invention stores a data transfer control program in each local memory of two or more processors, respectively. In the disk controller with which each processor performs the data transfer control program stored in self-local memory, respectively, and processes a data transfer job It is the cluster N from a cluster 1 (N) about two or more above-mentioned processors. A group division is carried out or more by two at N clusters to the natural number below [all] the number of processors. And the cluster limited mode in which distributed processing of a data transfer job is performed only between the processor groups belonging to the same cluster as the cluster to which the group division of the cluster group is carried out at two sets, the 1st set and the 2nd set, and a self-processor belongs. Cluster the mode in which it does not limit, in which distributed processing of a data transfer job is performed between the processor groups belonging to the cluster which belongs to the same group as the group to which the cluster to which the processor group and self-processor belonging to the same cluster as the cluster to which a self-processor belongs belong is formed, at the time of normal operation All clusters are made to belong to either the 1st set or the 2nd set, and distributed processing of the data transfer job is carried out in cluster the mode in which it does not limit, at the time of data transfer control program change operation First, processing of the data transfer job in cluster the mode in which it does not limit in a selection cluster is shifted to processing of the data transfer job in cluster limited mode, choosing one cluster and continuing distributed processing of the data transfer job in cluster the mode in which it does not limit in a non-choosing cluster. If shift is completed, registration of the new data transfer job in the processor group of a selection cluster will be stopped. The data transfer control program stored in each local memory of the processor group of a selection cluster if the data transfer job processed by the processor group of a selection cluster is lost is changed into a new data transfer control program, respectively. If change of the data transfer control program of all the processors of a selection cluster is completed, while resuming registration of the new data transfer job in the processor group of a selection cluster, and processing of a data transfer job After changing affiliation of a selection cluster into another side from either the 1st set or the 2nd set, processing of the data transfer job in the cluster limited mode in a selection cluster is returned to processing of the data transfer job in cluster the mode in which it does not limit. The non-stopped program change method of the disk controller characterized by repeating this in order about all clusters, and returning to normal operation is offered.

[0009] In the 6th viewpoint, this invention offers the disk controller which enforces the non-stopped program change method by the 4th viewpoint of the above, or the non-stopped program change method by the 5th viewpoint of the above.

[0010] [Function] By the non-stopped program change method of the multiprocessor system by the 1st viewpoint of the above The group division of the processor group is carried out at two or more clusters, and distributed processing of the job is usually carried out by all processors regardless of a cluster. At the time of program change First, it shifts to the mode which carries out distributed processing of the job only between the processors of the same cluster, and the processor of the predetermined cluster in which distributed processing is possible. Choosing one cluster and continuing processing by the non-choosing cluster, if it shifts, stop processing by the selection cluster and the program of the processor of a selection cluster is changed. After resuming processing of the program after change, repeating this about all clusters and completing change of a program and resumption of processing about all clusters, it is made to return to the usual operation which carries out distributed processing of the job by all processors regardless of a cluster. A program can be changed performing distributed processing between the processors of a different cluster by this without [without it stops operation of a multiprocessor system completely, and] the program of a different viewpoint of the above, since the above-mentioned non-stopped program change method can be enforced suitably, when changing a program, it is not necessary to stop operation completely.

[0011] By the non-stopped program change method of the disk controller by the 4th viewpoint of the above The group division of the processor group is carried out at two or more clusters, and distributed processing of the data transfer control job is usually carried out by all processors regardless of a cluster, at the time of data transfer control program change First, it shifts to the mode which carries out distributed processing of the data transfer control job only between the processors of the same cluster. Choosing one cluster and continuing processing by the non-choosing cluster, if it shifts, stop processing by the selection cluster and the data transfer control program of the processor of a selection cluster is changed. After resuming processing after change, repeating this about all clusters and completing change of a data transfer control program, and resumption of processing about all clusters, it is made to return to the usual operation which carries out distributed processing of the data transfer control job by all processors regardless of a cluster. Thereby, a data transfer control program can be changed, without [without it stops operation of a disk controller completely, and] the data transfer control program of a different version interfering.

[0012] In the multiprocessor system by the 3rd viewpoint of the above, since the above-mentioned non-stopped program change method can be enforced suitably, when changing a program, it is not necessary to stop operation completely.

[0013] By the non-stopped program change method of the disk controller by the 5th viewpoint of the above The group division of the processor group is carried out at two or more clusters, and distributed processing of the data transfer control job is usually carried out by all processors regardless of a cluster, at the time of data transfer control program change First, it shifts to the mode which carries out distributed processing of the data transfer control job only between the processors of the same cluster. Choosing one cluster and continuing processing by the non-choosing cluster, if it shifts, stop processing by the selection cluster and the data transfer control program of the processor of a selection cluster is changed. After resuming processing after change, repeating this about all clusters and completing change of a data transfer control program, and resumption of processing about all clusters, it is made to return to the usual operation which carries out distributed processing of the data transfer control job by all processors regardless of a cluster. Thereby, a data transfer control program can be changed, without [without it stops operation of a disk controller completely, and] the data transfer control program of a different version interfering.

[0014] By the non-stopped program change method of the disk controller by the 6th viewpoint of the above The group division of the processor group is carried out at two or more clusters, and distributed processing of the data transfer control job is usually carried out by all processors regardless of a cluster, at the time of data transfer control program change First, it shifts to the mode which carries out distributed processing of the data transfer control job only between the processors of the same cluster, and the processor of the predetermined cluster in which distributed processing is possible. Choosing one cluster and continuing processing by the non-choosing cluster, if it shifts, stop processing by the selection cluster and the data transfer control program of the processor of a selection cluster is changed. After resuming processing after change, repeating this about all clusters and completing change of a data transfer control program, and resumption of processing about all clusters, it is made to return to the usual operation which carries out distributed processing of the data transfer control job by all processors regardless of a cluster. A data transfer control program can be changed performing distributed processing between the processors of a different cluster by this without [without it stops operation of a disk controller completely, and] the data transfer control program of a different version interfering.

[0015] In the disk controller by the 6th viewpoint of the above, since the above-mentioned non-

stopped data transfer control program change method can be enforced suitably, when changing a data transfer control program, it is not necessary to stop operation completely.

[0016] Hereafter, the example of this invention is explained with reference to drawing. In addition, thereby, this invention is not limited.

[0017] – Example 1– drawing 1 is the block diagram of the disk controller 200 of the example 1 of this invention. This disk controller 200 possesses the processors 211, 212, 213, 214, 221, 222, 223, and 224 in which each has local memory LM, the service processor 215, 225 which has the drive 216, 226 for control program storing, respectively, and the shared memory 230 which has a cache memory 231 and a control memory 232, and is constituted.

[0018] The data transfer control program and the maintenance support program are stored in the local memory LM of eight aforementioned processors 211, 212, 213, 214, 221, 222, 223, and 224. The aforementioned data transfer control program is performed by each aforementioned processors 211, 212, 213, 214, 221, 222, 223, and 224, and data transfer processing is performed (that is, a disk controller 200 is a multiprocessor system). Moreover, the aforementioned maintenance support program is performed and non–stopped maintenance processing is performed. Processors 211, 212, 221 and 222 contain the channel CPU connected to channel controller CC among the aforementioned processor groups. Then, these are called channel side processor. Moreover, processors 213, 214, 221, 222, 223 and 224 include the drive CPU connected to the disk drive (only henceforth a drive) D0. Then, these are called drive side processor. Moreover, the group division of the aforementioned processor group is carried out at two clusters. Processors 211, 212, 213 and 214 are “clusters 1”, and processors 221, 222, 223 and 224 are “clusters 2”. The aforementioned service processor 215, 225 is connected to the maintenance terminal 240. Moreover, a service processor 215 belongs above “cluster 1”, and the service processor 225 belongs above “cluster 2.”

[0019] Whenever the aforementioned control memory 232 is shown in drawing 2, it comes, and it stores the cluster allocation information 500. The number of each processor is stored in the processor number area of this cluster allocation information 500 (this example 211, 212, 213, 214, 221, 222, 223, and 224). Moreover, the cluster number of the cluster to which a corresponding processor belongs is stored in affiliation cluster number area (in this example, processors 211, 212, 213 and 214 are cluster numbers “1”, and processors 221, 222, 223 and 224 are cluster numbers “2”). Moreover, the information which shows the function of each processor in distributed processing of a data transfer job is stored in processor attribute area (in this example, processors 211, 212, 221 and 222 are “Channels CPU”, and processors 213, 214, 223 and 224 are “Drives CPU”).

[0020] Moreover, it comes, whenever it is shown in drawing 3, and the execution cluster directions information 600 is stored in the aforementioned control memory 232. “1 (cluster mode in which it does not limit)” is stored in the execution cluster directions area of this execution cluster directions information 600 in the state where either the cluster 1 or a cluster 2 can be used for data transfer, and “2 (cluster limited mode)” is stored in it in the state where either a cluster 1 or the cluster 2 can be used for data transfer.

[0021] Moreover, it comes, whenever it is shown in drawing 4 created corresponding to all drives linked to a disk controller 200, and the I/O job execution control information 800 is stored in the aforementioned control memory 232. The command classification (a lead or light) of an access command, the drive number for access (it fixes every I/O job execution control information 800), and the cylinder number for access, a track number and a record number are stored in the access command demand area of this I/O job execution control information 800. Moreover, the information (“if it is an execute permission “1”, if execution is improper “0”) each processors 211, 212, 213, 214, 221, 222, 223, and 224 indicate it to be whether it is an execute permission for an access command is stored in execute–permission processor specification bit map area. Moreover, the information (it is “3”, if you have no execution demand, it is “0” and drive CPU execution waiting, it is “1” and channel CPU execution waiting and “2” and a certain processing are [be / it] under execution) which shows the running state of the access command to the drive concerned to I/O demand running state area is stored.

[0022] Furthermore, it comes, whenever it is shown in drawing 5, and the cluster mode–of–operation directions information 700 is stored in the aforementioned control memory 232. A cluster number (“a cluster 1”, “cluster 2”) is stored in the cluster number area of this cluster mode–of–operation directions information 700. Moreover, either of “3 (maintenance mode of operation)” which shows that the data transfer control program of a version with which “2 (pause mode)” or the cluster 1 which shows the mode of operation which does not receive “1 (normal mode of operation)” or the access command which shows the usual mode of operation differs from a cluster 2 is executed is stored in mode–of–operation area.

[0023] Next, data transfer control processing of a disk controller 200 is explained. If an access command is passed to channel controller CC from a host computer (illustration ellipsis), channel controller CC will pass the access command to any one of the channel side processors 211, 212, 221, 222 of the cluster 1 (210) of a disk controller 200, or a cluster 2 (220). Here, an access command presupposes that it is the lead demand to drive D0, and presupposes that the access command was passed to the channel side processor 211 of a cluster 1 (210). [0024] The channel side processor (211) to which the access command was passed It restricts, when “0 (with no execution demand)” is set to the I/O demand running state area of the I/O job execution control information 800 (drawing 4) corresponding to drive D0. “3 (under execution)” is set to I/O demand running state area, each information on an access command is stored in access command demand area, and “1 (waiting for drive CPU execution)” is set to I/O demand running state area. Next, with reference to the execution cluster directions information 600 (drawing 3), the aforementioned channel side processor (211) sets “1 (execution is possible)” to the execution propriety field corresponding to the drive side processor number (213, 214, 223, 224) of the execute–permission processor specification bit map area of the I/O job execution control information 800 (drawing 4). “1 (execution is possible)” is set only to the execution cluster (area is “2 (cluster limited mode)”. “1 (execution is possible)” is set to the execution propriety field corresponding to the drive side processor number (213, 214) belonging to the same cluster (here cluster 1) as the self–processor (here channel side processor 211) of the execute–permission processor specification bit map area of the I/O job execution control information 800 (drawing 4).

[0025] Arbitrary one out of drive side processors (213, 214, 223, 224) (here) the drive side processor 223 — carrying out, if “1 (waiting for drive CPU execution)” is set with reference to the I/O demand running state area of the I/O job execution control information 800 (drawing 4) (cluster mode in which it does not limit) “1 (execution is possible)” is set to the execution cluster directions area is “2 (cluster limited mode)”. The execution propriety field corresponding to the self–processor number (223) of the execute–permission processor specification bit map area of the I/O job execution control information 800 (drawing 4) concerned. And with reference to the I/O demand running state area of the I/O job execution control information 800 (drawing 3), when execution cluster directions area is “1 (cluster mode in which it does not limit)”, “1 (execution is possible)” is set to the execution propriety field corresponding to the channel side processor number (221, 222) of the execute–permission processor specification bit map area of the cache memory 232 of a shared memory 230, and “2 (waiting for channel CPU execution)” is set to the I/O demand running state area of the I/O job execution control information 800 (drawing 4) concerned. And with reference to the execution cluster directions information 600 (drawing 3), when execution cluster directions area is “1 (cluster mode in which it does not limit)”, “1 (execution is possible)” is set to the execution cluster directions area is “2 (cluster limited mode)”, “1 (execution is possible)” is set only to the execution propriety field corresponding to the channel side processor number (221, 222) of the execute–permission processor specification bit map area of the aforementioned I/O job execution control information 800 (drawing 4). On the other hand, when execution cluster directions area is “2 (cluster limited mode)”, “1 (execution is possible)” is set only to the execution propriety field corresponding to the channel side processor number (221, 222) belonging to the same cluster (cluster 2) as the self–processor (drive side processor 223) of the execute–permission processor specification bit map area of the aforementioned I/O job execution control information 800 (drawing 4).

[0026] Arbitrary one out of channel side processors (211, 212, 221, 222) (here) the channel side processor 221 — carrying out, if “2 (waiting for channel CPU execution)” is set with reference to the I/O demand running state area of the I/O job execution control information 800 (drawing 3)

4.) It restricts, when "1 (execution is possible)" is set to the execution propriety field corresponding to the self-processor number (221) of execute permission processor specification bit map area. "3 (under execution)" is set to I/O demand running state area, the data with which a cache memory 232 corresponds are transmitted to channel controller CC, and '0 (with no execution demand)" is set to I/O demand running state area. The above is operation at the time of the lead of data transfer control processing. It is the same also at the time of light, and explanation is omitted. Next, non-stopped maintenance processing (control program change processing) of a disk controller 20 is explained. Drawing 6 is the main flow view of non-stopped maintenance processing (control program change processing). A customer engineer stores the data transfer control program of a new version in the drive 218,226 for control program storing of a service processor 215,225 through the maintenance terminal 240 (Step 110). Next, a customer engineer directs a control program change processing start to a service processor 215 through the maintenance terminal 240 (Step 120). A service processor 215 sets "2 (cluster limited mode)" as the execution cluster directions information 600 (Drawing 3) (Step 125). Next, a service processor 215 performs control program change processing of a cluster 1 (Step 130). Control program change processing of this cluster 1 is explained in full detail with reference to drawing 7 later. While performing control program change processing about the processor (211,212,213,214) of a cluster 1, the processor (221,222,223,224) of a cluster 2 is continuing execution of the data transfer control program of an old version.

[0027] Next, a service processor 215 receives service-processor 225, and requires execution of control program change processing of a cluster 2 (Step 140). A service processor 225 performs control program change processing of a cluster 2 (Step 145). Control program change processing of this cluster 2 is explained in full detail with reference to drawing 8 later. While performing control program change processing about the processor (221,222,223,224) of a cluster 2, the processor (211,212,213,214) of a cluster 1 is performing the data transfer control program of a new version. If it checks that control program change processing of a cluster 2 has been completed (Step 150), a service processor 215 will set "1 (normal mode of operation)" as the mode of operation of each cluster of the cluster mode-of-operation directions information 700 (Drawing 5), and will set "1 (cluster mode in which it does not limit)" as the execution cluster directions information 600 (Drawing 3) (Step 155). And the message of completion of control program change processing is displayed on the maintenance terminal 240 (Step 160).

[0028] Drawing 7 is the detailed flow view of control program change processing (130) of the above-mentioned cluster 1. It checks that the processor (211,212,213,214) of a cluster 1 does not have the job (cluster non-limiting job) which is "1 (execution is possible)" simultaneously to the processor of a cluster 1, and the processor of a cluster 2 with reference to the execute-permission processor specification bit map area of the I/O job execution control information 800 (Drawing 4) on all drives (D0, D1, --) (Step 310). Since the execution cluster directions area of the execution cluster directions information 600 on drawing 3 is set to "2 (cluster limited mode)" by Step 125 of drawing 6, even if a new access command is passed to one of the channel side processors (211,212,221,222) from channel controller CC, "1 (execution is possible)" is set only to one processor of a cluster 1 or a cluster 2 by the execute permission processor specification bit map area of the I/O job execution control information 800 (Drawing 4). Therefore, cluster a non-limiting job will be extinguished someday. This disappearance is checked.

[0029] Next, a service processor 215 sets the mode of operation of the cluster 1 of the cluster mode-of-operation directions information 700 on drawing 5 as "2 (pause mode)" (Step 320). The channel side processor (211,212) of a cluster 1 notifies not receiving a new access command to channel controller CC, when the mode of operation of the cluster 1 of the cluster mode-of-operation directions information 700 is set as "2 (pause mode)" (Step 320). Thereby, the new job which the processor of a cluster 1 should perform will not be registered henceforth. Next, it checks that the processor (211,212,213,214) of a cluster 1 does not have the job (cluster limited job) which is "1 (execution is possible)" to the processor of a cluster 1 with reference to the execute permission processor specification bit map area of the I/O job execution control information 800 (Drawing 4) on all drives (D0, D1, --) (Step 325). And the completion of a pause of a self-processor is notified to a service processor 215 (Step 330).

[0030] Next, a service processor 215 sets the mode of operation of the cluster 1 of the cluster mode-of-operation directions information 700 on drawing 5 as "3 (maintenance mode of operation)" (Step 335). Next, it is required for a service processor 215 that the processor (211,212,213,214) which detected that the mode of operation of a cluster 1 was set as the maintenance mode of operation should transmit the data transfer control program of a new version (Step 340). According to this, a service processor 215 transmits the data transfer control program of a new version to the processor (211,212,213,214) of a cluster 1, and stores it in each local memory LM (Step 345). The channel side processor (211,212) of a cluster 1 will notify resuming registration of a new access command to channel controller CC, if it checks that the data transfer control program of a new version has been stored in the local memory LM of all the processors (211,212,213,214) of a cluster 1 (Step 350).

[0031] Drawing 8 is the detailed flow view of control program change processing (145) of the above-mentioned cluster 2. In detail, the job which exists at the time of the start of control program change processing (145) of a cluster 2 is only a cluster limited job, and cluster a non-limiting job does not exist. First, a service processor 225 sets the mode of operation of the cluster 2 of the cluster mode-of-operation directions information 700 on drawing 5 as "2 (pause mode)" (Step 405). The channel side processor (221,222) of a cluster 2 notifies not receiving a new access command to channel controller CC, when the mode of operation of the cluster 2 of the cluster mode-of-operation directions information 700 is set as "2 (pause mode)" (Step 415). Thereby, the new job which the processor of a cluster 2 should perform will not be registered henceforth. Next, it checks that the processor (221,222,223,224) of a cluster 2 does not have the job (cluster limited job) which is "1 (execution is possible)" to the processor of a cluster 2 with reference to the execute-permission processor specification bit map area of the I/O job execution control information 800 (Drawing 4) on all drives (D0, D1, --) (Step 420). And the completion of a pause of a self-processor is notified to a service processor 225 (Step 425). [0032] Next, a service processor 225 sets the mode of operation of the cluster 2 of the cluster mode-of-operation directions information 700 on drawing 5 as "3 (maintenance mode of operation)" (Step 426). Next, it is required for a service processor 225 that the processor (221,222,223,224) which detected that the mode of operation of a cluster 2 was set as the maintenance mode of operation should transmit the data transfer control program of a new version (Step 427). According to this, a service processor 225 transmits the data transfer control program of a new version to the processor (221,222,223,224) of a cluster 2, and stores it in each local memory LM (Step 430). The channel side processor (221,222) of a cluster 2 will notify resuming registration of a new access command to channel controller CC, if it checks that the data transfer control program of a new version has been stored in the local memory LM of all the processors (221,222,223,224) of a cluster 2 (Step 440). [0033] According to the above disk controller 200, a data transfer control program can be upgraded, without [without it stops a processor group (221,222,223,224) completely, and] making the processor comrade who performs the data transfer control program with which versions differ interfere.

[0034] - Example 2- drawing 9 is the block diagram of disk controller 200 of the example 2 of this invention. This disk controller 200 is the same composition as the disk controller 200 of an example 1 except for the point that cluster division of a processor group is four.

[0035] When cluster division is set or more to three, two or more combination of the cluster in which distributed processing of a job is possible can be defined between processor groups. Whenever the combination of the cluster in which this distributed processing is possible is shown in drawing 10, it comes, and it creates and stores the distributed-processing possible cluster combination information 100 in a control memory 232. In the distributed-processing possible cluster combination information 100 on drawing 10, if distributed processing of a job is possible for the processor group of the cluster number of a horizontal axis, and the processor group of the cluster number of a horizontal axis, "O.K." will be set as the field common to these cluster numbers. If distributed processing of a job is not possible, "NG" will be set as the field common to these cluster numbers. Usually, when the processor group of the cluster number of a vertical axis and the processor group of the cluster number of a horizontal axis are performing the data

transfer control program of the same version, "O.K." is set up, and "NG" is set up when the data transfer control program of the same version cannot be performed. For example, when the processor group of all clusters is performing the data transfer control program of the same version, as shown in drawing 10, "O.K." is set as all the fields. On the other hand, only the processor group (2|11|213) of a cluster 1 has started execution of the data transfer control program of a new version, and when execution of the data transfer control program of an old version is being continued, the processor (212, 214, 221|223,222|224) of a cluster 2 - a cluster 4 sets "NG" to the field corresponding to a cluster 1, as shown in drawing 11, and sets "O.K." to the field common to a cluster 2 - a cluster 4. Thus, since the version of the data transfer control program which the processor group of each cluster performs changes at the time of maintenance operation, it changes dynamically the set point of the distributed-processing possible cluster combination information 100 according to it.

[0036] And when the execution cluster appointed area of the execution cluster specification information 800 on drawing 3 of each processor is "2 (cluster limited mode)", The inside of the execution propriety field of the execute-permission processor specification bit map area of the I/O job execution control information 800 on drawing 4, "O.K." is set up to the cluster to which a self-processor belongs for the above-mentioned distributed-processing possible cluster combination information 100, and also "1 (execution is possible)" is set only to the execution propriety field corresponding to the processor belonging to a cluster.

[0037] According to the above disk controller 200', a data transfer control program can be upgraded in a finer cluster unit. Moreover, the version of a data transfer control program can carry out distributed processing of the job between the same processors.

[0038] The drawing 11 equivalent view at the time of carrying out the group division of the processor group at five clusters is shown in drawing 12. Program change processing of the processor of a cluster 1 and a cluster 2 ends this distributed-processing possible cluster combination information 100; it is the middle of performing program change processing of a cluster 3 and the content at the time of program change processing of the processor of a cluster 4 and a cluster 5 not being started yet is shown. That is, since the processor of a cluster 1 and a cluster 2 is performing the new program, distributed processing is possible for while and the cluster 1 and the cluster 2 have become a group. Moreover, since the processor of a cluster 4 and a cluster 5 is performing the old program, distributed processing is possible for a cluster 4 and a cluster 5, and also they have become the group of a way. Moreover, since a cluster 3 is [program / be / it] under change, with other clusters, distributed processing of it has not become possible.

[0039] - Example 3 - drawing 13 is the block diagram of disk controller 200" of the example 3 of this invention. This disk controller 200" is the composition of having connected the processor (211|212|213|214), the service processor (215), and the shared memory (230) by bus 201. In this case, if the processor (211|212|213|214) linked to a bus 201 is divided into a suitable cluster, a data transfer control program can be upgraded by the same maintenance procedure as the above-mentioned example 1 or the above-mentioned example 2.

[0040] [Effect of the Invention] According to the non-stopped program change method of the multiprocessor system of this invention, a program can be changed, without [without it stops operation of a multiprocessor system completely, and] the program of a different version interfering.

[0041] It becomes unnecessary to carry out a planned halt of the operation for change of a program according to the multiprocessor system of this invention.

[0042] According to the non-stopped program change method of the disk controller of this invention, a data transfer control program can be changed, without [without it stops operation of a disk controller completely, and] the data transfer control program of a different version interfering.

[0043] It becomes unnecessary to carry out a planned halt of the operation for change of a data transfer control program according to the disk controller of this invention.

[Translation done.]

* NOTICES *

Japan Patent Office is not responsible for any damage caused by the use of this translation.

- 1.This document has been translated by computer. So the translation may not reflect the original precisely.
- 2.**** shows the word which can not be translated.
- 3.In the drawings, any words are not translated.

DESCRIPTION OF DRAWINGS**[Brief Description of the Drawings]**

[Drawing 1] It is the block diagram of the disk controller of the example 1 of this invention.
 [Drawing 2] It is the block diagram of cluster allocation information.
 [Drawing 3] It is the block diagram of execution cluster directions information.
 [Drawing 4] It is the block diagram of I/O Job execution control information.
 [Drawing 5] It is the block diagram of cluster mode-of-operation directions information.
 [Drawing 6] It is the main flow view of non-stopped maintenance processing (control program change processing).
 [Drawing 7] It is the detailed flow view of control program change processing of a cluster 1.
 [Drawing 8] It is the detailed flow view of control program change processing of a cluster 2.
 [Drawing 9] It is the block diagram of the disk controller of the example 2 of this invention.
 [Drawing 10] It is the block diagram of distributed-processing possible cluster combination information.
 [Drawing 11] It is another block diagram of distributed-processing possible cluster combination information.
 [Drawing 12] It is still more nearly another block diagram of distributed-processing possible cluster combination information.
 [Drawing 13] It is the block diagram of the disk controller of the example 3 of this invention.

[Descriptions of Notations]

100,100' Distributed-processing possible cluster combination information
 200,200',200'' Disk controller
 201 [] Bus
 211,212,221,222 Channel side processor
 213,214,223,224 Drive side processor
 215,225 Service processor
 230 [] Shared Memory
 232 [] Control Memory
 240 [] Maintenance Terminal
 500 [] Cluster Quota Information
 600 [] Execution Cluster Directions Information
 800 [] I/O Job Execution Control Information
 700 [] Cluster Mode-of-Operation Directions Information
 CC Channel controller
 D0, D1 Disk drive

[Translation done.]

(19)日本国特許庁 (JP)

(12) 公開特許公報 (A)

(11)特許出願公開番号

特開平7-306844

(43)公開日 平成7年(1995)11月21日

(51)Int.Cl.⁶
G 0 6 F 15/177
9/06

識別記号 庁内整理番号
5 4 0 F 7230-5B

F I

技術表示箇所

G 0 6 F 15/ 16 4 2 0 S

審査請求 未請求 請求項の数12 O L (全 17 頁)

(21)出願番号 特願平6-99705

(22)出願日 平成6年(1994)5月13日

(71)出願人 000005108

株式会社日立製作所

東京都千代田区神田駿河台四丁目6番地

(72)発明者 村田 智洋

神奈川県川崎市麻生区王禅寺1099番地 株式会社日立製作所システム開発研究所内

(72)発明者 仁田 満秋

神奈川県小田原市国府津2880番地 株式会社日立製作所ストレージシステム事業部内

(72)発明者 栗原 謙三

神奈川県川崎市麻生区王禅寺1099番地 株式会社日立製作所システム開発研究所内

(74)代理人 弁理士 有近 純志郎

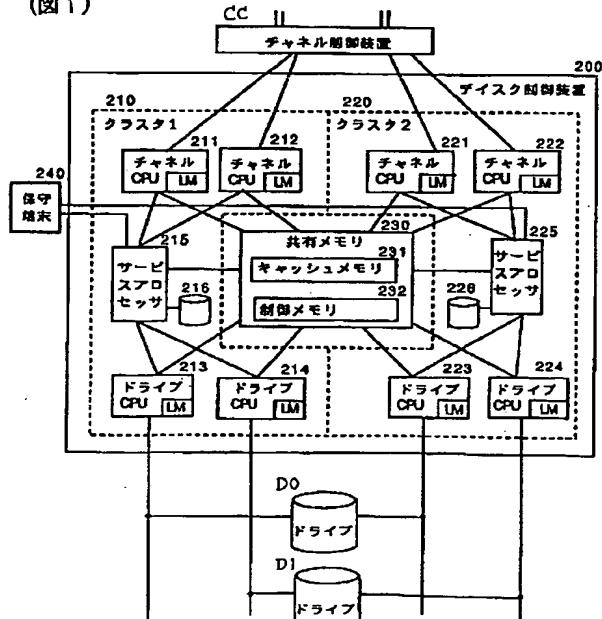
(54)【発明の名称】 マルチプロセッサシステムの無停止プログラム変更方法およびマルチプロセッサシステムおよびディスク制御装置の無停止プログラム変更方法およびディスク制御装置

(57)【要約】 (修正有)

【目的】 プロセッサ群を全面停止させることなく、プロセッサ群が実行するプログラムを変更する。

【構成】 プロセッサ群を2個のクラスタ(クラスタ1, クラスタ2)にグループ分けし、通常はクラスタに関係なく全てのプロセッサでデータ転送制御ジョブを分散処理し、データ転送制御プログラム変更時は、まず、同一クラスタのプロセッサ間でのみジョブを分散処理するモードに移行し、移行できたらクラスタ2での処理を継続しながらクラスタ1での処理を休止してクラスタ1のプロセッサの各ローカルメモリLMのデータ転送制御プログラムを変更し、変更後に処理を再開し、次に、クラスタ1での処理を継続しながらクラスタ2での処理を休止してクラスタ2のプロセッサの各ローカルメモリLMのデータ転送制御プログラムを変更し、変更後に処理を再開し、その後、クラスタに関係なく全てのプロセッサでジョブを分散処理する通常の動作に復帰させる。

(図1)



【特許請求の範囲】

【請求項1】 複数のプロセッサの各ローカルメモリにそれぞれプログラムを格納し、各プロセッサが自ローカルメモリに格納したプログラムをそれぞれ実行してジョブを処理するマルチプロセッサシステムにおいて、上記複数のプロセッサをクラスタ1からクラスタN（Nは、2以上で全プロセッサ数以下の自然数）までのN個のクラスタにグループ分けし、且つ、自プロセッサが属するクラスタと同一クラスタに属するプロセッサ群の間でのみジョブの分散処理を行なうクラスタ限定モードと全てのプロセッサの間でジョブの分散処理を行なうクラスタ非限定モードとを設け、通常動作時は、クラスタ非限定モードでジョブを分散処理し、

プログラム変更動作時は、まず、クラスタ非限定モードでのジョブの処理からクラスタ限定モードでのジョブの処理に移行し、移行が完了したら一つのクラスタを選択し、非選択クラスタのプロセッサ群でのジョブの処理および新たなジョブの受け付けを継続しつつ選択クラスタのプロセッサ群での新たなジョブの受け付けを停止し、選択クラスタのプロセッサ群で処理されるジョブが無くなったら選択クラスタのプロセッサ群の各ローカルメモリに格納されているプログラムを新たなプログラムにそれぞれ変更し、選択クラスタの全てのプロセッサのプログラムの変更を完了したら選択クラスタのプロセッサ群での新たなジョブの受け付けおよびジョブの処理を再開し、これを全てのクラスタについて順に繰り返した後、クラスタ限定モードでのジョブの処理をクラスタ非限定モードでのジョブの処理に戻して通常動作に戻ることを特徴とするマルチプロセッサシステムの無停止プログラム変更方法。

【請求項2】 複数のプロセッサの各ローカルメモリにそれぞれプログラムを格納し、各プロセッサが自ローカルメモリに格納したプログラムをそれぞれ実行してジョブを処理するマルチプロセッサシステムにおいて、上記複数のプロセッサをクラスタ1からクラスタN（Nは、2以上で全プロセッサ数以下の自然数）までのN個のクラスタにグループ分けし、且つ、クラスタ群を第1組と第2組の2個の集合にグループ分けし、且つ、自プロセッサが属するクラスタと同一クラスタに属するプロセッサ群の間でのみジョブの分散処理を行なうクラスタ限定モードと、自プロセッサが属するクラスタと同一クラスタに属するプロセッサ群および自プロセッサが属するクラスタが所属する組と同一組に所属するクラスタに属するプロセッサ群の間でジョブの分散処理を行なうクラスタ非限定モードとを設け、

通常動作時は、全てのクラスタを第1組または第2組の一方に所属させ、クラスタ非限定モードでジョブを分散処理し、

プログラム変更動作時は、まず、一つのクラスタを選択

し、非選択クラスタにおけるクラスタ非限定モードでのジョブの分散処理を継続しつつ選択クラスタにおけるクラスタ非限定モードでのジョブの処理をクラスタ限定モードでのジョブの処理へと移行し、移行が完了したら選択クラスタのプロセッサ群での新たなジョブの受け付けを停止し、選択クラスタのプロセッサ群で処理されるジョブが無くなったら選択クラスタのプロセッサ群の各ローカルメモリに格納されているプログラムを新たなプログラムにそれぞれ変更し、選択クラスタの全てのプロセッサのプログラムの変更を完了したら選択クラスタのプロセッサ群での新たなジョブの受け付けおよびジョブの処理を再開すると共に、選択クラスタの所属を第1組または第2組の一方から他方に変更した上で選択クラスタにおけるクラスタ限定モードでのジョブの処理をクラスタ非限定モードでのジョブの処理に戻し、これを全てのクラスタについて順に繰り返して通常動作に戻ることを特徴とするマルチプロセッサシステムの無停止プログラム変更方法。

【請求項3】 複数のプロセッサの各ローカルメモリにそれぞれプログラムを格納し、各プロセッサが自ローカルメモリに格納したプログラムをそれぞれ実行してジョブを処理するマルチプロセッサシステムにおいて、上記複数のプロセッサをクラスタ1からクラスタN（Nは、2以上で全プロセッサ数以下の自然数）までのN個のクラスタにグループ分けするクラスタ割り当て手段と、

自プロセッサが属するクラスタと同一クラスタに属するプロセッサ群の間でのみジョブの分散処理を行うクラスタ限定モードか、全てのプロセッサの間でジョブの分散処理を行なうクラスタ非限定モードか、のいずれかで分散処理を行うよう制御する分散処理モード制御手段と、通常動作時は、前記クラスタ非限定モードによりジョブの分散処理を行なわせ、プログラム変更動作時は、まず、クラスタ非限定モードでのジョブの処理からクラスタ限定モードでのジョブの処理に移行させ、移行が完了したら一つのクラスタを選択し、非選択クラスタのプロセッサ群でのジョブの処理および新たなジョブの受け付けを継続させつつ選択クラスタのプロセッサ群での新たなジョブの受け付けを停止させ、選択クラスタのプロセッサ群で処理されるジョブが無くなったら選択クラスタのプロセッサ群の各ローカルメモリに格納されているプログラムを新たなプログラムにそれぞれ変更し、選択クラスタの全てのプロセッサのプログラムの変更を完了したら選択クラスタのプロセッサ群での新たなジョブの受け付けおよびジョブの処理を再開させ、これを全てのクラスタについて順に繰り返した後、クラスタ限定モードでのジョブの処理をクラスタ非限定モードでのジョブの処理に移行させ通常動作に戻らせるプログラム変更処理手段とを具備したことを特徴とするマルチプロセッサシステム。

【請求項4】 複数のプロセッサの各ローカルメモリにそれぞれプログラムを格納し、各プロセッサが自ローカルメモリに格納したプログラムをそれぞれ実行してジョブを処理するマルチプロセッサシステムにおいて、上記複数のプロセッサをクラスタ1からクラスタN（Nは、2以上で全プロセッサ数以下の自然数）までのN個のクラスタにグループ分けするクラスタ割り当て手段と、

クラスタ群を第1組と第2組の2個の集合にグループ分けする分散処理可能クラスタ組合せ手段と、

自プロセッサが属するクラスタと同一クラスタに属するプロセッサ群の間でのみジョブの分散処理を行なうクラスタ限定モードか、自プロセッサが属するクラスタと同一クラスタに属するプロセッサ群および自プロセッサが属するクラスタが所属する組と同一組に所属するクラスタに属するプロセッサ群の間でジョブの分散処理を行なうクラスタ非限定モードか、のいずれかで分散処理を行うよう制御する分散処理モード制御手段と、

通常動作時は、全てのクラスタを第1組または第2組の一方に所属させ、クラスタ非限定モードでジョブを分散処理させ、プログラム変更動作時は、まず、一つのクラスタを選択し、非選択クラスタにおけるクラスタ非限定モードでのジョブの分散処理を継続させつつ選択クラスタにおけるクラスタ非限定モードでのジョブの処理をクラスタ限定モードでのジョブの処理へと移行させ、移行が完了したら選択クラスタのプロセッサ群での新たなジョブの受け付けを停止させ、選択クラスタのプロセッサ群で処理されるジョブが無くなったら選択クラスタのプロセッサ群の各ローカルメモリに格納されているプログラムを新たにプログラムにそれぞれ変更し、選択クラスタの全てのプロセッサのプログラムの変更を完了したら選択クラスタのプロセッサ群での新たなジョブの受け付けおよびジョブの処理を再開させると共に、選択クラスタの所属を第1組または第2組の一方から他方に変更させた上で選択クラスタにおけるクラスタ限定モードでのジョブの処理をクラスタ非限定モードでのジョブの処理に移行させ、これを全てのクラスタについて順に繰り返して通常動作に戻らせるプログラム変更処理手段とを具備したことを特徴とするマルチプロセッサシステム。

【請求項5】 複数のプロセッサの各ローカルメモリにそれぞれプログラムを格納し、各プロセッサが自ローカルメモリに格納したプログラムをそれぞれ実行してジョブを処理するマルチプロセッサシステムにおいて、上記複数のプロセッサを2以上のクラスタにグループ分けするために各プロセッサがいずれのクラスタに属しているかを示すクラスタ割当情報を記憶するクラスタ割当情報記憶手段と、

ローカルメモリに格納したプログラムに従ってプロセッサがジョブの処理を行なうノーマル動作モードか、クラスタを閉塞するためにプロセッサがジョブの処理を休止する休止モードか、ローカルメモリに格納されたプログラムを新たにプログラムに変更する保守動作モードか、のいずれの動作モードにするかをクラスタ単位で示すクラスタ動作モード指示情報を記憶する動作モード指示情報記憶手段と、

る休止モードか、ローカルメモリに格納されたプログラムを新たにプログラムに変更する保守動作モードか、のいずれの動作モードにするかをクラスタ単位で示すクラスタ動作モード指示情報を記憶する動作モード指示情報記憶手段と、

自プロセッサが属するクラスタと同じクラスタに属するプロセッサとのみジョブの分散処理を行なうクラスタ限定モードか、クラスタに拘り無く任意のプロセッサ間でジョブの分散処理を行なうクラスタ非限定モードか、のいずれの分散処理モードでジョブを処理するかを示す実行クラスタ指示情報を記憶する実行クラスタ指示情報記憶手段と、

プログラムの変更開始を指示するプログラム変更開始指示手段と、

プログラムの変更開始が指示されたとき、分散処理モードをクラスタ限定モードに設定し、任意のクラスタを一つ選び、そのクラスタの動作モードを休止モードに設定し、当該クラスタに属するプロセッサが全て休止状態になった時点で当該クラスタの動作モードを保守動作モードに設定し、当該クラスタに属する全てのプロセッサのローカルメモリに格納されているプログラムを新たにプログラムに変更し、これを全クラスタについて繰り返し、その完了時点で全てのクラスタの動作モードをノーマル動作モードに設定し、分散処理モードをクラスタ非限定モードに設定するプログラム変更処理手段と、を具備したことを特徴とするマルチプロセッサシステム。

【請求項6】 複数のプロセッサの各ローカルメモリにそれぞれプログラムを格納し、各プロセッサが自ローカルメモリに格納したプログラムをそれぞれ実行してジョブを処理するマルチプロセッサシステムにおいて、上記複数のプロセッサを2以上のクラスタにグループ分けするために各プロセッサがいずれのクラスタに属しているかを示すクラスタ割当情報を記憶するクラスタ割当情報記憶手段と、

ローカルメモリに格納したプログラムに従ってプロセッサがジョブの処理を行なうノーマル動作モードか、クラスタを閉塞するためにプロセッサがジョブの処理を休止する休止モードか、ローカルメモリに格納されたプログラムを新たにプログラムに変更する保守動作モードか、のいずれの動作モードにするかをクラスタ単位で示すクラスタ動作モード指示情報を記憶する動作モード指示情報記憶手段と、

自プロセッサが属するクラスタと同じクラスタに属するプロセッサとのみジョブの分散処理を行なうクラスタ限定モードか、自プロセッサが属するクラスタと同じクラスタに属するプロセッサおよび自プロセッサとの間でジョブの分散処理が可能な他クラスタのプロセッサでジョブの分散処理を行なうクラスタ非限定モードか、のいずれの分散処理モードでジョブを処理するかを示す実行クラスタ指示情報を記憶する実行クラスタ指示情報記憶手

段と、

プログラムの変更開始を指示するプログラム変更開始指示手段と、

プログラムの変更開始が指示されたとき、任意のクラスタを一つ選び、そのクラスタの分散処理モードをクラスタ限定モードに設定し、当該クラスタの動作モードを休止モードに設定し、当該クラスタに属するプロセッサが全て休止状態になった時点で当該クラスタの動作モードを保守動作モードに設定し、当該クラスタに属する全てのプロセッサのローカルメモリに格納されているプログラムを新たなプログラムに変更し、当該クラスタの分散処理モードをクラスタ非限定モードに設定し、これを全クラスタについて繰り返し、その完了時点で全てのクラスタの動作モードをノーマル動作モードに設定するプログラム変更処理手段と、を具備したことを特徴とするマルチプロセッサシステム。

【請求項7】 複数のプロセッサの各ローカルメモリにそれぞれデータ転送制御プログラムを格納し、各プロセッサが自ローカルメモリに格納したデータ転送制御プログラムをそれぞれ実行してディスクドライブに対するデータ転送ジョブを処理するディスク制御装置において、上記複数のプロセッサをクラスタ1からクラスタN（Nは、2以上で全プロセッサ数以下の自然数）までのN個のクラスタにグループ分けし、且つ、自プロセッサが属するクラスタと同一クラスタに属するプロセッサ群の間でのみデータ転送ジョブの分散処理を行なうクラスタ限定モードと全てのプロセッサの間でデータ転送ジョブの分散処理を行なうクラスタ非限定モードとを設け、通常動作時は、クラスタ非限定モードでデータ転送ジョブを分散処理し、

データ転送制御プログラム変更動作時は、まず、クラスタ非限定モードでのデータ転送ジョブの処理からクラスタ限定モードでのデータ転送ジョブの処理に移行し、移行が完了したら一つのクラスタを選択し、非選択クラスタのプロセッサ群でのデータ転送ジョブの処理および新たなデータ転送ジョブの受け付けを継続しつつ選択クラスタのプロセッサ群での新たなデータ転送ジョブの受け付けを停止し、選択クラスタのプロセッサ群で処理されるデータ転送ジョブが無くなったら選択クラスタのプロセッサ群の各ローカルメモリに格納されているデータ転送制御プログラムを新たなデータ転送制御プログラムにそれぞれ変更し、選択クラスタの全てのプロセッサのデータ転送制御プログラムの変更を完了したら選択クラスタのプロセッサ群での新たなデータ転送ジョブの受け付けおよびデータ転送ジョブの処理を再開し、これを全てのクラスタについて順に繰り返した後、クラスタ限定モードでのデータ転送ジョブの処理をクラスタ非限定モードでのデータ転送ジョブの処理に戻して通常動作に戻ることを特徴とするディスク制御装置の無停止プログラム変更方法。

【請求項8】 複数のプロセッサの各ローカルメモリにそれぞれデータ転送制御プログラムを格納し、各プロセッサが自ローカルメモリに格納したデータ転送制御プログラムをそれぞれ実行してデータ転送ジョブを処理するディスク制御装置において、

上記複数のプロセッサをクラスタ1からクラスタN（Nは、2以上で全プロセッサ数以下の自然数）までのN個のクラスタにグループ分けし、且つ、クラスタ群を第1組と第2組の2個の集合にグループ分けし、且つ、自プロセッサが属するクラスタと同一クラスタに属するプロセッサ群の間でのみデータ転送ジョブの分散処理を行なうクラスタ限定モードと、自プロセッサが属するクラスタと同一クラスタに属するプロセッサ群および自プロセッサが属するクラスタが所属する組と同一組に所属するクラスタに属するプロセッサ群の間でデータ転送ジョブの分散処理を行なうクラスタ非限定モードとを設け、通常動作時は、全てのクラスタを第1組または第2組の一方に所属させ、クラスタ非限定モードでデータ転送ジョブを分散処理し、データ転送制御プログラム変更動作時は、まず、一つのクラスタを選択し、非選択クラスタにおけるクラスタ非限定モードでのデータ転送ジョブの分散処理を継続しつつ選択クラスタにおけるクラスタ非限定モードでのデータ転送ジョブの処理をクラスタ限定モードでのデータ転送ジョブの処理へと移行し、移行が完了したら選択クラスタのプロセッサ群での新たなデータ転送ジョブの受け付けを停止し、選択クラスタのプロセッサ群で処理されるデータ転送ジョブが無くなったら選択クラスタのプロセッサ群の各ローカルメモリに格納されているデータ転送制御プログラムを新たなデータ転送制御プログラムにそれぞれ変更し、選択クラスタの全てのプロセッサのデータ転送制御プログラムの変更を完了したら選択クラスタのプロセッサ群での新たなデータ転送ジョブの受け付けおよびデータ転送ジョブの処理を再開すると共に、選択クラスタの所属を第1組または第2組の一方から他方に変更した上で選択クラスタにおけるクラスタ限定モードでのデータ転送ジョブの処理をクラスタ非限定モードでのデータ転送ジョブの処理に戻し、これを全てのクラスタについて順に繰り返して通常動作に戻ることを特徴とするディスク制御装置の無停止プログラム変更方法。

【請求項9】 複数のプロセッサの各ローカルメモリにそれぞれデータ転送制御プログラムを格納し、各プロセッサが自ローカルメモリに格納したデータ転送制御プログラムをそれぞれ実行してデータ転送ジョブを処理するディスク制御装置において、

上記複数のプロセッサをクラスタ1からクラスタN（Nは、2以上で全プロセッサ数以下の自然数）までのN個のクラスタにグループ分けするクラスタ割り当て手段と、

自プロセッサが属するクラスタと同一クラスタに属する

プロセッサ群の間でのみデータ転送ジョブの分散処理を行うクラスタ限定モードか、全てのプロセッサの間でデータ転送ジョブの分散処理を行なうクラスタ非限定モードか、のいずれかで分散処理を行うよう制御する分散処理モード制御手段と、

通常動作時は、前記クラスタ非限定モードによりデータ転送ジョブの分散処理を行なわせ、データ転送制御プログラム変更動作時は、まず、クラスタ非限定モードでのデータ転送ジョブの処理からクラスタ限定モードでのデータ転送ジョブの処理に移行させ、移行が完了したら一つのクラスタを選択し、非選択クラスタのプロセッサ群でのデータ転送ジョブの処理および新たなデータ転送ジョブの受け付けを継続させつつ選択クラスタのプロセッサ群での新たなデータ転送ジョブの受け付けを停止させ、選択クラスタのプロセッサ群で処理されるデータ転送ジョブが無くなったら選択クラスタのプロセッサ群の各ローカルメモリに格納されているデータ転送制御プログラムを新たなデータ転送制御プログラムにそれぞれ変更し、選択クラスタの全てのプロセッサのデータ転送制御プログラムの変更を完了したら選択クラスタのプロセッサ群での新たなデータ転送ジョブの受け付けおよびデータ転送ジョブの処理を再開させ、これを全てのクラスタについて順に繰り返した後、クラスタ限定モードでのデータ転送ジョブの処理をクラスタ非限定モードでのデータ転送ジョブの処理に移行させ通常動作に戻らせるデータ転送制御プログラム変更処理手段とを具備したこととを特徴とするディスク制御装置。

【請求項10】 複数のプロセッサの各ローカルメモリにそれぞれデータ転送制御プログラムを格納し、各プロセッサが自ローカルメモリに格納したデータ転送制御プログラムをそれぞれ実行してデータ転送ジョブを処理するディスク制御装置において、

上記複数のプロセッサをクラスタ1からクラスタN（Nは、2以上で全プロセッサ数以下の自然数）までのN個のクラスタにグループ分けするクラスタ割り当て手段と、

クラスタ群を第1組と第2組の2個の集合にグループ分けする分散処理可能クラスタ組合せ手段と、

自プロセッサが属するクラスタと同一クラスタに属するプロセッサ群の間でのみデータ転送ジョブの分散処理を行なうクラスタ限定モードか、自プロセッサが属するクラスタと同一クラスタに属するプロセッサ群および自プロセッサが属するクラスタが所属する組と同一組に所属するクラスタに属するプロセッサ群の間でデータ転送ジョブの分散処理を行なうクラスタ非限定モードか、のいずれかで分散処理を行うよう制御する分散処理モード制御手段と、

通常動作時は、全てのクラスタを第1組または第2組の一方に所属させ、クラスタ非限定モードでデータ転送ジョブを分散処理させ、データ転送制御プログラム変更動

作時は、まず、一つのクラスタを選択し、非選択クラスタにおけるクラスタ非限定モードでのデータ転送ジョブの分散処理を継続させつつ選択クラスタにおけるクラスタ非限定モードでのデータ転送ジョブの処理をクラスタ限定モードでのデータ転送ジョブの処理へと移行させ、移行が完了したら選択クラスタのプロセッサ群での新たなデータ転送ジョブの受け付けを停止させ、選択クラスタのプロセッサ群で処理されるデータ転送ジョブが無くなったら選択クラスタのプロセッサ群の各ローカルメモリに格納されているデータ転送制御プログラムを新たなデータ転送制御プログラムにそれぞれ変更し、選択クラスタの全てのプロセッサのデータ転送制御プログラムの変更を完了したら選択クラスタのプロセッサ群での新たなデータ転送ジョブの受け付けおよびデータ転送ジョブの処理を再開させると共に、選択クラスタの所属を第1組または第2組の一方から他方に変更させた上で選択クラスタにおけるクラスタ限定モードでのデータ転送ジョブの処理をクラスタ非限定モードでのデータ転送ジョブの処理に移行させ、これを全てのクラスタについて順に繰り返して通常動作に戻らせるデータ転送制御プログラム変更処理手段とを具備したことを特徴とするディスク制御装置。

【請求項11】 複数のプロセッサの各ローカルメモリにそれぞれデータ転送制御プログラムを格納し、各プロセッサが自ローカルメモリに格納したデータ転送制御プログラムをそれぞれ実行してデータ転送ジョブを処理するディスク制御装置において、

上記複数のプロセッサを2以上のクラスタにグループ分けするために各プロセッサがいずれのクラスタに属しているかを示すクラスタ割り当て情報を記憶するクラスタ割り当て情報記憶手段と、

ローカルメモリに格納したデータ転送制御プログラムに従ってプロセッサがデータ転送ジョブの処理を行なうノーマル動作モードか、クラスタを閉塞するためにプロセッサがデータ転送ジョブの処理を休止する休止モードか、ローカルメモリに格納されたデータ転送制御プログラムを新たなデータ転送制御プログラムに変更する保守動作モードか、のいずれの動作モードにするかをクラスタ単位で示すクラスタ動作モード指示情報を記憶する動作モード指示情報記憶手段と、

自プロセッサが属するクラスタと同じクラスタに属するプロセッサとのみデータ転送ジョブの分散処理を行なうクラスタ限定モードか、クラスタに拘り無く任意のプロセッサ間でデータ転送ジョブの分散処理を行なうクラスタ非限定モードか、のいずれの分散処理モードでデータ転送ジョブを処理するかを示す実行クラスタ指示情報を記憶する実行クラスタ指示情報記憶手段と、

データ転送制御プログラムの変更開始を指示するデータ転送制御プログラム変更開始指示手段と、

データ転送制御プログラムの変更開始が指示されたと

き、分散処理モードをクラスタ限定モードに設定し、任意のクラスタを一つ選び、そのクラスタの動作モードを休止モードに設定し、当該クラスタに属するプロセッサが全て休止状態になった時点で当該クラスタの動作モードを保守動作モードに設定し、当該クラスタに属する全てのプロセッサのローカルメモリに格納されているデータ転送制御プログラムを新たなデータ転送制御プログラムに変更し、これを全クラスタについて繰り返し、その完了時点で全てのクラスタの動作モードをノーマル動作モードに設定し、分散処理モードをクラスタ非限定モードに設定するデータ転送制御プログラム変更処理手段と、を具備したことを特徴とするディスク制御装置。

【請求項12】複数のプロセッサの各ローカルメモリにそれぞれデータ転送制御プログラムを格納し、各プロセッサが自ローカルメモリに格納したデータ転送制御プログラムをそれぞれ実行してデータ転送ジョブを処理するディスク制御装置において、

上記複数のプロセッサを2以上のクラスタにグループ分けするために各プロセッサがいずれのクラスタに属しているかを示すクラスタ割当て情報を記憶するクラスタ割当て情報記憶手段と、

ローカルメモリに格納したデータ転送制御プログラムに従ってプロセッサがデータ転送ジョブの処理を行うノーマル動作モードか、クラスタを閉塞するためにプロセッサがデータ転送ジョブの処理を休止する休止モードか、ローカルメモリに格納されたデータ転送制御プログラムを新たにデータ転送制御プログラムに変更する保守動作モードか、のいずれの動作モードにするかをクラスタ単位で示すクラスタ動作モード指示情報を記憶する動作モード指示情報記憶手段と、

自プロセッサが属するクラスタと同じクラスタに属するプロセッサとのみデータ転送ジョブの分散処理を行なうクラスタ限定モードか、自プロセッサが属するクラスタと同じクラスタに属するプロセッサおよび自プロセッサとの間でデータ転送ジョブの分散処理が可能な他クラスタのプロセッサでデータ転送ジョブの分散処理を行なうクラスタ非限定モードか、のいずれの分散処理モードでデータ転送ジョブを処理するかを示す実行クラスタ指示情報を記憶する実行クラスタ指示情報記憶手段と、データ転送制御プログラムの変更開始を指示するデータ転送制御プログラム変更開始指示手段と、

データ転送制御プログラムの変更開始が指示されたとき、任意のクラスタを一つ選び、そのクラスタの分散処理モードをクラスタ限定モードに設定し、当該クラスタの動作モードを休止モードに設定し、当該クラスタに属するプロセッサが全て休止状態になった時点で当該クラスタの動作モードを保守動作モードに設定し、当該クラスタに属する全てのプロセッサのローカルメモリに格納されているデータ転送制御プログラムを新たにデータ転送制御プログラムに変更し、当該クラスタの分散処理モ

ードをクラスタ非限定モードに設定し、これを全クラスタについて繰り返し、その完了時点で全てのクラスタの動作モードをノーマル動作モードに設定するデータ転送制御プログラム変更処理手段と、を具備したことを特徴とするディスク制御装置。

【発明の詳細な説明】

【0001】

【産業上の利用分野】本発明は、マルチプロセッサシステムの無停止プログラム変更方法およびマルチプロセッサシステムおよびディスク制御装置の無停止プログラム変更方法およびディスク制御装置に関する。さらに詳しくは、プロセッサ群を全面停止させることなく、プロセッサ群が実行するプログラムを変更することが出来るマルチプロセッサシステムの無停止プログラム変更方法およびマルチプロセッサシステムおよびディスク制御装置の無停止プログラム変更方法およびディスク制御装置に関する。

【0002】

【従来の技術】主系プロセッサと従系プロセッサとを設け、主系プロセッサで行なっていた処理を従系プロセッサで引き継ぐバックアップシステム切り替えを行なった後、主系プロセッサを停止させて保守する無停止保守方法が例えば特開昭56-52460号公報に記載されている。

【0003】

【発明が解決しようとする課題】上記従来の無停止保守方法によれば、処理を全面停止させることなく、主系プロセッサに属するシステム部分を保守することが可能である。しかし、主系プロセッサで行なっていた処理を従系プロセッサで引き継ぐために、主系プロセッサと従系プロセッサがそれぞれ実行するプログラムが同じバージョンであることが必要であり、プログラムをバージョンアップする場合には上記無停止保守方法を適用できない問題点がある。すなわち、従来の技術では、プログラムをバージョンアップする場合には、プロセッサ群を全面停止させなければならない問題点がある。そこで、本発明の目的は、プロセッサ群を全面停止させることなく、プロセッサ群が実行するプログラムを変更することが出来るマルチプロセッサシステムの無停止プログラム変更方法およびマルチプロセッサシステムおよびディスク制御装置の無停止プログラム変更方法およびディスク制御装置を提供することにある。

【0004】

【課題を解決するための手段】第1の観点では、本発明は、複数のプロセッサの各ローカルメモリにそれぞれプログラムを格納し、各プロセッサが自ローカルメモリに格納したプログラムをそれぞれ実行してジョブを処理するマルチプロセッサシステムにおいて、上記複数のプロセッサをクラスタ1からクラスタN（Nは、2以上で全プロセッサ数以下の自然数）までのN個のクラスタにグ

ループ分けし、且つ、自プロセッサが属するクラスタと同一クラスタに属するプロセッサ群の間でのみジョブの分散処理を行なうクラスタ限定モードと全てのプロセッサの間でジョブの分散処理を行なうクラスタ非限定モードとを設け、通常動作時は、クラスタ非限定モードでジョブを分散処理し、プログラム変更動作時は、まず、クラスタ非限定モードでのジョブの処理からクラスタ限定モードでのジョブの処理に移行し、移行が完了したら一つのクラスタを選択し、非選択クラスタのプロセッサ群でのジョブの処理および新たなジョブの受け付けを継続しつつ選択クラスタのプロセッサ群での新たなジョブの受け付けを停止し、選択クラスタのプロセッサ群で処理されるジョブが無くなったら選択クラスタのプロセッサ群の各ローカルメモリに格納されているプログラムを新たなプログラムにそれぞれ変更し、選択クラスタの全てのプロセッサのプログラムの変更を完了したら選択クラスタのプロセッサ群での新たなジョブの受け付けおよびジョブの処理を再開し、これを全てのクラスタについて順に繰り返した後、クラスタ限定モードでのジョブの処理をクラスタ非限定モードでのジョブの処理に戻して通常動作に戻ることを特徴とするマルチプロセッサシステムの無停止プログラム変更方法を提供する。

【0005】第2の観点では、本発明は、複数のプロセッサの各ローカルメモリにそれぞれプログラムを格納し、各プロセッサが自ローカルメモリに格納したプログラムをそれぞれ実行してジョブを処理するマルチプロセッサシステムにおいて、上記複数のプロセッサをクラスタ1からクラスタN（Nは、2以上で全プロセッサ数以下の自然数）までのN個のクラスタにグループ分けし、且つ、クラスタ群を第1組と第2組の2個の集合にグループ分けし、且つ、自プロセッサが属するクラスタと同一クラスタに属するプロセッサ群の間でのみジョブの分散処理を行なうクラスタ限定モードと、自プロセッサが属するクラスタと同一クラスタに属するプロセッサ群および自プロセッサが属するクラスタが所属する組と同一組に所属するクラスタに属するプロセッサ群の間でジョブの分散処理を行なうクラスタ非限定モードとを設け、通常動作時は、全てのクラスタを第1組または第2組の一方に所属させ、クラスタ非限定モードでジョブを分散処理し、プログラム変更動作時は、まず、一つのクラスタを選択し、非選択クラスタにおけるクラスタ非限定モードでのジョブの分散処理を継続しつつ選択クラスタにおけるクラスタ非限定モードでのジョブの処理へと移行し、移行が完了したら選択クラスタのプロセッサ群での新たなジョブの受け付けを停止し、選択クラスタのプロセッサ群で処理されるジョブが無くなったら選択クラスタのプロセッサ群の各ローカルメモリに格納されているプログラムを新たなプログラムにそれぞれ変更し、選択クラスタの全てのプロセッサのデータ転送制御プログラムの変更を完了したら選択クラスタのプロセッサ群での新たなデータ転送ジョブの受け付けおよびデータ転送ジョブの処理を再開し、これを全てのクラスタについて順に繰り返した後、クラスタ限定モードでのデータ転送ジョブの処理をクラスタ非限定モードでのデータ転送ジョブの処理に戻して通常動作に戻ることを特徴とするディスク制御装置の無停止プログラム変更方法を提供する。

【0006】第3の観点では、本発明は、上記第1の観点による無停止プログラム変更方法または上記第2の観点による無停止プログラム変更方法を実施するマルチプロセッサシステムを提供する。

【0007】第4の観点では、本発明は、複数のプロセッサの各ローカルメモリにそれぞれデータ転送制御プログラムを格納し、各プロセッサが自ローカルメモリに格納したデータ転送制御プログラムをそれぞれ実行してディスクドライブに対するデータ転送ジョブを処理するディスク制御装置において、上記複数のプロセッサをクラスタ1からクラスタN（Nは、2以上で全プロセッサ数以下の自然数）までのN個のクラスタにグループ分けし、且つ、自プロセッサが属するクラスタと同一クラスタに属するプロセッサ群の間でのみデータ転送ジョブの分散処理を行なうクラスタ限定モードと全てのプロセッサの間でデータ転送ジョブの分散処理を行なうクラスタ非限定モードとを設け、通常動作時は、クラスタ非限定モードでデータ転送ジョブを分散処理し、データ転送制御プログラム変更動作時は、まず、クラスタ非限定モードでのデータ転送ジョブの処理からクラスタ限定モードでのデータ転送ジョブの処理に移行し、移行が完了したら一つのクラスタを選択し、非選択クラスタのプロセッサ群でのデータ転送ジョブの処理および新たなデータ転送ジョブの受け付けを継続しつつ選択クラスタのプロセッサ群での新たなデータ転送ジョブの受け付けを停止し、選択クラスタのプロセッサ群で処理されるデータ転送ジョブが無くなったら選択クラスタのプロセッサ群の各ローカルメモリに格納されているデータ転送制御プログラムを新たなデータ転送制御プログラムにそれぞれ変更し、選択クラスタの全てのプロセッサのデータ転送制御プログラムの変更を完了したら選択クラスタのプロセッサ群での新たなデータ転送ジョブの受け付けおよびデータ転送ジョブの処理を再開し、これを全てのクラスタについて順に繰り返した後、クラスタ限定モードでのデータ転送ジョブの処理をクラスタ非限定モードでのデータ転送ジョブの処理に戻して通常動作に戻ることを特徴とするディスク制御装置の無停止プログラム変更方法を提供する。

【0008】第5の観点では、本発明は、複数のプロセッサの各ローカルメモリにそれぞれデータ転送制御プログラムを格納し、各プロセッサが自ローカルメモリに格納したデータ転送制御プログラムをそれぞれ実行してデ

ータ転送ジョブを処理するディスク制御装置において、上記複数のプロセッサをクラスタ1からクラスタN（Nは、2以上で全プロセッサ数以下の自然数）までのN個のクラスタにグループ分けし、且つ、クラスタ群を第1組と第2組の2個の集合にグループ分けし、且つ、自プロセッサが属するクラスタと同一クラスタに属するプロセッサ群の間でのみデータ転送ジョブの分散処理を行なうクラスタ限定モードと、自プロセッサが属するクラスタと同一クラスタに属するプロセッサ群および自プロセッサが属するクラスタが所属する組と同一組に所属するクラスタに属するプロセッサ群の間でデータ転送ジョブの分散処理を行なうクラスタ非限定モードとを設け、通常動作時は、全てのクラスタを第1組または第2組の一方に所属させ、クラスタ非限定モードでデータ転送ジョブを分散処理し、データ転送制御プログラム変更動作時は、まず、一つのクラスタを選択し、非選択クラスタにおけるクラスタ非限定モードでのデータ転送ジョブの分散処理を継続しつつ選択クラスタにおけるクラスタ非限定モードでのデータ転送ジョブの処理をクラスタ限定モードでのデータ転送ジョブの処理へと移行し、移行が完了したら選択クラスタのプロセッサ群での新たなデータ転送ジョブの受け付けを停止し、選択クラスタのプロセッサ群で処理されるデータ転送ジョブが無くなったら選択クラスタのプロセッサ群の各ローカルメモリに格納されているデータ転送制御プログラムを新たなデータ転送制御プログラムにそれぞれ変更し、選択クラスタの全てのプロセッサのデータ転送制御プログラムの変更を完了したら選択クラスタのプロセッサ群での新たなデータ転送ジョブの受け付けおよびデータ転送ジョブの処理を再開すると共に、選択クラスタの所属を第1組または第2組の一方から他方に変更した上で選択クラスタにおけるクラスタ限定モードでのデータ転送ジョブの処理をクラスタ非限定モードでのデータ転送ジョブの処理に戻し、これを全てのクラスタについて順に繰り返して通常動作に戻ることを特徴とするディスク制御装置の無停止プログラム変更方法を提供する。

【0009】第6の観点では、本発明は、上記第4の観点による無停止プログラム変更方法または上記第5の観点による無停止プログラム変更方法を実施するディスク制御装置を提供する。

【0010】

【作用】上記第1の観点によるマルチプロセッサシステムの無停止プログラム変更方法では、プロセッサ群を2以上のクラスタにグループ分けし、通常はクラスタに関係なく全てのプロセッサでジョブを分散処理し、プログラム変更時は、まず、同一クラスタのプロセッサ間でのみジョブを分散処理するモードに移行し、移行できたら一つのクラスタを選択し、非選択クラスタでの処理を継続しながら選択クラスタでの処理を休止して選択クラスタのプロセッサのプログラムを変更し、変更後に処理を

再開し、これを全てのクラスタについて繰り返し、全てのクラスタについてプログラムの変更と処理の再開とが完了した後、クラスタに関係なく全てのプロセッサでジョブを分散処理する通常の動作に復帰させる。これにより、マルチプロセッサシステムの動作を全面停止されることなく、且つ、異なるバージョンのプログラムが干渉することなく、プログラムを変更することが出来る。

【0011】上記第2の観点によるマルチプロセッサシステムの無停止プログラム変更方法では、プロセッサ群を2以上のクラスタにグループ分けし、通常はクラスタに関係なく全てのプロセッサでジョブを分散処理し、プログラム変更時は、まず、同一クラスタのプロセッサおよび分散処理可能な所定のクラスタのプロセッサ間でのみジョブを分散処理するモードに移行し、移行できたら一つのクラスタを選択し、非選択クラスタでの処理を継続しながら選択クラスタでの処理を休止して選択クラスタのプロセッサのプログラムを変更し、変更後に処理を再開し、これを全てのクラスタについて繰り返し、全てのクラスタについてプログラムの変更と処理の再開とが完了した後、クラスタに関係なく全てのプロセッサでジョブを分散処理する通常の動作に復帰させる。これにより、マルチプロセッサシステムの動作を全面停止されることなく、且つ、異なるバージョンのプログラムが干渉することなく、且つ、異なるクラスタのプロセッサ間での分散処理を行いながら、プログラムを変更することが出来る。

【0012】上記第3の観点によるマルチプロセッサシステムでは、上記無停止プログラム変更方法を好適に実施可能であるから、プログラムを変更するときに動作を全面停止させずに済む。

【0013】上記第4の観点によるディスク制御装置の無停止プログラム変更方法では、プロセッサ群を2以上のクラスタにグループ分けし、通常はクラスタに関係なく全てのプロセッサでデータ転送制御ジョブを分散処理し、データ転送制御プログラム変更時は、まず、同一クラスタのプロセッサ間でのみデータ転送制御ジョブを分散処理するモードに移行し、移行できたら一つのクラスタを選択し、非選択クラスタでの処理を継続しながら選択クラスタでの処理を休止して選択クラスタのプロセッサのデータ転送制御プログラムを変更し、変更後に処理を再開し、これを全てのクラスタについて繰り返し、全てのクラスタについてデータ転送制御プログラムの変更と処理の再開とが完了した後、クラスタに関係なく全てのプロセッサでデータ転送制御ジョブを分散処理する通常の動作に復帰させる。これにより、ディスク制御装置の動作を全面停止されることなく、且つ、異なるバージョンのデータ転送制御プログラムが干渉することなく、データ転送制御プログラムを変更することが出来る。

【0014】上記第5の観点によるディスク制御装置の無停止プログラム変更方法では、プロセッサ群を2以上

のクラスタにグループ分けし、通常はクラスタに関係なく全てのプロセッサでデータ転送制御ジョブを分散処理し、データ転送制御プログラム変更時は、まず、同一クラスタのプロセッサおよび分散処理可能な所定のクラスタのプロセッサ間でのみデータ転送制御ジョブを分散処理するモードに移行し、移行できたら一つのクラスタを選択し、非選択クラスタでの処理を休止して選択クラスタのプロセッサのデータ転送制御プログラムを変更し、変更後に処理を再開し、これを全てのクラスタについて繰り返し、全てのクラスタについてデータ転送制御プログラムの変更と処理の再開とが完了した後、クラスタに関係なく全てのプロセッサでデータ転送制御ジョブを分散処理する通常の動作に復帰させる。これにより、ディスク制御装置の動作を全面停止させることなく、且つ、異なるバージョンのデータ転送制御プログラムが干渉することなく、且つ、異なるクラスタのプロセッサ間での分散処理を行なながら、データ転送制御プログラムを変更することが出来る。

【0015】上記第6の観点によるディスク制御装置では、上記無停止データ転送制御プログラム変更方法を好適に実施可能であるから、データ転送制御プログラムを変更するときに動作を全面停止させずに済む。

【0016】

【実施例】以下、図を参照して本発明の実施例を説明する。なお、これにより本発明が限定されるものではない。

【0017】－実施例1－

図1は、本発明の実施例1のディスク制御装置200の構成図である。このディスク制御装置200は、各々がローカルメモリLMを有するプロセッサ211, 212, 213, 214, 221, 222, 223および224と、制御プログラム格納用ドライブ216, 226をそれぞれ有するサービスプロセッサ215, 225と、キャッシュメモリ231および制御メモリ232を有する共有メモリ230とを具備して構成されている。

【0018】前記8台のプロセッサ211, 212, 213, 214, 221, 222, 223および224のローカルメモリLMには、データ転送制御プログラムと保守支援プログラムとが格納されている。前記各プロセッサ211, 212, 213, 214, 221, 222, 223および224により前記データ転送制御プログラムが実行され、データ転送制御処理が行われる（すなわち、ディスク制御装置200はマルチプロセッサシステムである）。また、前記保守支援プログラムが実行され、無停止保守処理が行われる。前記プロセッサ群のうちプロセッサ211, 212, 221および222は、チャネル制御装置CCに接続されたチャネルCPUを含んでいる。そこで、これらをチャネル側プロセッサと呼ぶ。また、プロセッサ213, 214, 223およ

び224は、ディスクドライブ（以下、単にドライブという）D0に接続されたドライブCPUを含んでいる。そこで、これらをドライブ側プロセッサと呼ぶ。また、前記プロセッサ群は、2つのクラスタにグループ分けされている。プロセッサ211, 212, 213および214は“クラスタ1”であり、プロセッサ221, 222, 223および224は“クラスタ2”である。前記サービスプロセッサ215, 225は、保守末端240に接続されている。また、サービスプロセッサ215は前記“クラスタ1”に属し、サービスプロセッサ225は前記“クラスタ2”に属している。

【0019】前記制御メモリ232は、図2に示すときクラスタ割当て情報500を格納している。このクラスタ割当て情報500のプロセッサ番号エリアには、各プロセッサの番号が格納されている（本実施例では、211, 212, 213, 214, 221, 222, 223および224）。また、所属クラスタ番号エリアには、対応するプロセッサが属するクラスタのクラスタ番号が格納されている（本実施例では、プロセッサ211, 212, 213および214がクラスタ番号“1”であり、プロセッサ221, 222, 223および224がクラスタ番号“2”である）。また、プロセッサ属性エリアには、データ転送ジョブの分散処理における各プロセッサの機能を示す情報が格納されている（本実施例では、プロセッサ211, 212, 221および222が“チャネルCPU”であり、プロセッサ213, 214, 223および224が“ドライブCPU”である）。

【0020】また、前記制御メモリ232には、図3に示すとき実行クラスタ指示情報600を格納している。この実行クラスタ指示情報600の実行クラスタ指示エリアには、クラスタ1およびクラスタ2のいずれでもデータ転送に使用できる状態のときは“1（クラスタ非限定モード）”を格納し、クラスタ1またはクラスタ2のいずれか一方のみをデータ転送に使用できる状態のときは“2（クラスタ限定モード）”を格納する。

【0021】また、前記制御メモリ232には、ディスク制御装置200に接続する全てのドライブに対応して作成した図4に示すときI/Oジョブ実行制御情報800を格納している。このI/Oジョブ実行制御情報800のアクセスコマンド要求エリアには、アクセスコマンドのコマンド種別（リードまたはライト）と、アクセス対象のドライブ番号（I/Oジョブ実行制御情報800ごとに固定）と、アクセス対象のシリンド番号、トラック番号、レコード番号などを格納する。また、実行可能プロセッサ指定ビットマップエリアには、各プロセッサ211, 212, 213, 214, 221, 222, 223および224がアクセスコマンドを実行可能か否かを示す情報（実行可能なら“1”、実行不可なら“0”）を格納する。また、I/O要求実行状態エリア

には、当該ドライブに対するアクセスコマンドの実行状態を示す情報（実行要求無しなら“0”、ドライブCPU実行待ちなら“1”、チャネルCPU実行待ちなら“2”、何らかの処理を実行中なら“3”）を格納する。

【0022】さらに、前記制御メモリ232には、図5に示すときクラスタ動作モード指示情報700を格納している。このクラスタ動作モード指示情報700のクラスタ番号エリアには、クラスタ番号（“クラスタ1”，“クラスタ2”）を格納する。また、動作モードエリアには、通常の動作モードを示す“1（ノーマル動作モード）”またはアクセスコマンドを受け付けない動作モードを示す“2（休止モード）”またはクラスタ1とクラスタ2とが異なるバージョンのデータ転送制御プログラムを実行していることを示す“3（保守動作モード）”のいずれかを格納する。

【0023】次に、ディスク制御装置200のデータ転送制御処理を説明する。ホストコンピュータ（図省略）からチャネル制御装置CCにアクセスコマンドが渡されると、チャネル制御装置CCは、そのアクセスコマンドを、ディスク制御装置200のクラスタ1（210）またはクラスタ2（220）のチャネル側プロセッサ211，212，221，222のいずれか一つに渡す。ここでは、アクセスコマンドがドライブD0に対するリード要求であるとし、そのアクセスコマンドがクラスタ1（210）のチャネル側プロセッサ211に渡されたとする。

【0024】アクセスコマンドを渡されたチャネル側プロセッサ（211）は、ドライブD0に対応するI/Oジョブ実行制御情報800（図4）のI/O要求実行状態エリアに“0（実行要求無し）”がセットされている場合に限り、I/O要求実行状態エリアに“3（実行中）”をセットし、アクセスコマンド要求エリアにアクセスコマンドの各情報を格納し、I/O要求実行状態エリアに“1（ドライブCPU実行待ち）”をセットする。次に、前記チャネル側プロセッサ（211）は、実行クラスタ指示情報600（図3）を参照し、実行クラスタ指示エリアが“1（クラスタ非限定モード）”であるときは、I/Oジョブ実行制御情報800（図4）の実行可能プロセッサ指定ビットマップエリアのうちのドライブ側プロセッサ番号（213，214，223，224）に対応する実行可否フィールドに“1（実行可）”をセットする。一方、実行クラスタ指示エリアが“2（クラスタ限定モード）”であるときは、I/Oジョブ実行制御情報800（図4）の実行可能プロセッサ指定ビットマップエリアのうちの自プロセッサ（ここではチャネル側プロセッサ211）と同一のクラスタ（ここではクラスタ1）に属するドライブ側プロセッサ番号（213，214）に対応する実行可否フィールドのみに“1（実行可）”をセットする。

【0025】ドライブ側プロセッサ（213，214，223，224）のうちの任意の一つ（ここでは、ドライブ側プロセッサ223とする）は、I/Oジョブ実行制御情報800（図4）のI/O要求実行状態エリアを参照し、“1（ドライブCPU実行待ち）”がセットされていれば、当該I/Oジョブ実行制御情報800（図4）の実行可能プロセッサ指定ビットマップエリアの自プロセッサ番号（223）に対応する実行可否フィールドを参照し、そこに“1（実行可）”がセットされている場合に限り、I/O要求実行状態エリアに“3（実行中）”をセットし、アクセスコマンド要求エリアに格納されている情報（ドライブ番号、シリンド番号、トラック番号、レコード番号）に従ってデータを読み出し、共有メモリ230のキャッシュメモリ232に格納し、当該I/Oジョブ実行制御情報800（図4）のI/O要求実行状態エリアに“2（チャネルCPU実行待ち）”をセットする。そして、実行クラスタ指示情報600（図3）を参照し、実行クラスタ指示エリアが“1（クラスタ非限定モード）”であるときは、前記I/Oジョブ実行制御情報800（図4）の実行可能プロセッサ指定ビットマップエリアのうちのチャネル側プロセッサ番号（211，212，221，222）に対応する実行可否フィールドに“1（実行可）”をセットする。一方、実行クラスタ指示エリアが“2（クラスタ限定モード）”であるときは、前記I/Oジョブ実行制御情報800（図4）の実行可能プロセッサ指定ビットマップエリアのうちの自プロセッサ（ドライブ側プロセッサ223）と同一のクラスタ（クラスタ2）に属するチャネル側プロセッサ番号（221，222）に対応する実行可否フィールドのみに“1（実行可）”をセットする。

【0026】チャネル側プロセッサ（211，212，221，222）のうちの任意の一つ（ここでは、チャネル側プロセッサ221とする）は、I/Oジョブ実行制御情報800（図4）のI/O要求実行状態エリアを参照し、“2（チャネルCPU実行待ち）”がセットされていれば、実行可能プロセッサ指定ビットマップエリアの自プロセッサ番号（221）に対応する実行可否フィールドに“1（実行可）”がセットされている場合に限り、I/O要求実行状態エリアに“3（実行中）”をセットし、キャッシュメモリ232の該当するデータをチャネル制御装置CCに転送し、I/O要求実行状態エリアに“0（実行要求無し）”をセットする。以上がデータ転送制御のリード時の動作である。ライト時も同様であり、説明は省略する。次に、ディスク制御装置200の無停止保守処理（制御プログラム変更処理）を説明する。図6は、無停止保守処理（制御プログラム変更処理）のメインフロー図である。保守員は、保守端末240を介して、サービスプロセッサ215，225の制御プログラム格納用ドライブ216，226に新しいバージョンのデータ転送制御プログラムを格納する（ス

ステップ110)。次に、保守員は、保守端末240を介して、サービスプロセッサ215に制御プログラム変更処理開始を指示する(ステップ120)。サービスプロセッサ215は、実行クラスタ指示情報600(図3)に“2(クラスタ限定モード)”を設定する(ステップ125)。次に、サービスプロセッサ215は、クラスタ1の制御プログラム変更処理を実行する(ステップ130)。このクラスタ1の制御プログラム変更処理については、図7を参照して後で詳述する。クラスタ1のプロセッサ(211, 212, 213, 214)について制御プログラム変更処理を実行している間、クラスタ2のプロセッサ(221, 222, 223, 224)は、古いバージョンのデータ転送制御プログラムの実行を継続している。

【0027】 次に、サービスプロセッサ215は、サービスプロセッサ225に対して、クラスタ2の制御プログラム変更処理の実行を要求する（ステップ140）。サービスプロセッサ225は、クラスタ2の制御プログラム変更処理を実行する（ステップ145）。このクラスタ2の制御プログラム変更処理については、図8を参照して後で詳述する。クラスタ2のプロセッサ（221、222、223、224）について制御プログラム変更処理を実行している間、クラスタ1のプロセッサ（211、212、213、214）は、新しいバージョンのデータ転送制御プログラムを実行している。サービスプロセッサ215は、クラスタ2の制御プログラム変更処理が完了したことを確認すると（ステップ150）、クラスタ動作モード指示情報700（図5）の各クラスタの動作モードに“1（ノーマル動作モード）”を設定し、且つ、実行クラスタ指示情報600（図3）に“1（クラスタ非限定モード）”を設定する（ステップ155）。そして、保守端末240に、制御プログラム変更処理の完了のメッセージを表示する（ステップ160）。

【0028】図7は、上記クラスタ1の制御プログラム変更処理(130)の詳細フロー図である。クラスタ1のプロセッサ(211, 212, 213, 214)は、全てのドライブ(D0, D1, ...)のI/Oジョブ実行制御情報800(図4)の実行可能プロセッサ指定ビットマップエリアを参照し、クラスタ1のプロセッサおよびクラスタ2のプロセッサに対して同時に“1(実行可)”となるジョブ(クラスタ非限定ジョブ)がないことを確認する(ステップ310)。図6のステップ125により図3の実行クラスタ指示情報600の実行クラスタ指示エリアが“2(クラスタ限定モード)”とされているため、チャネル側プロセッサ(211, 212, 221, 222)の一つがチャネル制御装置CCから新たなアクセスコマンドを渡されても、I/Oジョブ実行制御情報800(図4)の実行可能プロセッサ指定ビットマップエリアには、クラスタ1またはクラスタ2

の一方のプロセッサにのみ“1（実行可）”がセットされる。従って、クラスタ非限定ジョブはいずれ消滅することになる。この消滅を確認する。

【0029】 次に、サービスプロセッサ215は、図5のクラスタ動作モード指示情報700のクラスタ1の動作モードを“2（休止モード）”に設定する（ステップ320）。クラスタ1のチャネル側プロセッサ（211，212）は、クラスタ動作モード指示情報700のクラスタ1の動作モードが“2（休止モード）”に設定されている場合、新規のアクセスコマンドの受け付けを行なわないことをチャネル制御装置CCに通知する（ステップ320）。これにより、以後、クラスタ1のプロセッサが実行すべき新たなジョブは登録されなくなる。次に、クラスタ1のプロセッサ（211，212，213，214）は、全てのドライブ（D0，D1，…）のI/Oジョブ実行制御情報800（図4）の実行可能プロセッサ指定ビットマップエリアを参照し、クラスタ1のプロセッサに対して“1（実行可）”となってるジョブ（クラスタ限定ジョブ）がないことを確認する（ステップ325）。そして、自プロセッサの休止完了をサービスプロセッサ215に通知する（ステップ330）。

【0030】 次に、サービスプロセッサ215は、図5のクラスタ動作モード指示情報700のクラスタ1の動作モードを“3（保守動作モード）”に設定する（ステップ335）。次に、クラスタ1の動作モードが保守動作モードに設定されたことを検知したプロセッサ（211, 212, 213, 214）は、新しいバージョンのデータ転送制御プログラムを転送するようにサービスプロセッサ215に要求する（ステップ340）。これに応じて、サービスプロセッサ215は、新しいバージョンのデータ転送制御プログラムをクラスタ1のプロセッサ（211, 212, 213, 214）に転送し、各々のローカルメモリLMに格納する（ステップ345）。クラスタ1のチャネル側プロセッサ（211, 212）は、新しいバージョンのデータ転送制御プログラムがクラスタ1の全てのプロセッサ（211, 212, 213, 214）のローカルメモリLMに格納されたことを確認すると、新規のアクセスコマンドの受け付けを再開することをチャネル制御装置CCに通知する（ステップ350）。

【0031】図8は、上記クラスタ2の制御プログラム変更処理(145)の詳細フロー図である。なお、クラスタ2の制御プログラム変更処理(145)の開始時に存在するジョブはクラスタ限定ジョブのみであり、クラスタ非限定ジョブは存在しない。まず、サービスプロセッサ225は、図5のクラスタ動作モード指示情報700のクラスタ2の動作モードを“2(休止モード)”に設定する(ステップ405)。クラスタ2のチャネル側プロセッサ(221, 222)は、クラスタ動作モード指示情報700のクラスタ2の動作モードが“2(休止

モード)”に設定されている場合、新規のアクセスコマンドの受付けを行なわないことをチャネル制御装置CCに通知する(ステップ415)。これにより、以後、クラスタ2のプロセッサが実行すべき新たなジョブは登録されなくなる。次に、クラスタ2のプロセッサ(221, 222, 223, 224)は、全てのドライブ(D0, D1, …)のI/Oジョブ実行制御情報800(図4)の実行可能プロセッサ指定ビットマップエリアを参考し、クラスタ2のプロセッサに対して“1(実行可)”となるジョブ(クラスタ限定ジョブ)がないことを確認する(ステップ420)。そして、自プロセッサの休止完了をサービスプロセッサ225に通知する(ステップ425)。

【0032】次に、サービスプロセッサ225は、図5のクラスタ動作モード指示情報700のクラスタ2の動作モードを“3(保守動作モード)”に設定する(ステップ426)。次に、クラスタ2の動作モードが保守動作モードに設定されたことを検知したプロセッサ(221, 222, 223, 224)は、新しいバージョンのデータ転送制御プログラムを転送するようにサービスプロセッサ225に要求する(ステップ427)。これに応じて、サービスプロセッサ225は、新しいバージョンのデータ転送制御プログラムをクラスタ2のプロセッサ(221, 222, 223, 224)に転送し、各々のローカルメモリLMに格納する(ステップ430)。クラスタ2のチャネル側プロセッサ(221, 222)は、新しいバージョンのデータ転送制御プログラムがクラスタ2の全てのプロセッサ(221, 222, 223, 224)のローカルメモリLMに格納されたことを確認すると、新規のアクセスコマンドの受付けを再開することをチャネル制御装置CCに通知する(ステップ440)。

【0033】以上のディスク制御装置200によれば、プロセッサ群(211, 212, 213, 214, 221, 222, 223および224)を全面停止させることなく、かつ、バージョンの異なるデータ転送制御プログラムを実行するプロセッサ同志を干渉させることなく、データ転送制御プログラムをバージョンアップすることが出来る。

【0034】-実施例2-

図9は、本発明の実施例2のディスク制御装置200'の構成図である。このディスク制御装置200'は、プロセッサ群のクラスタ分割が4つになっている点を除いて、実施例1のディスク制御装置200と同じ構成である。

【0035】クラスタ分割を3つ以上にした場合、プロセッサ群の間でジョブの分散処理が可能なクラスタの組合せを複数定めが出来る。この分散処理可能なクラスタの組合せは、図10に示すごとき分散処理可能クラスタ組合せ情報100を制御メモリ232に作成して

格納する。図10の分散処理可能クラスタ組合せ情報100において、縦軸のクラスタ番号のプロセッサ群と横軸のクラスタ番号のプロセッサ群とがジョブの分散処理可能であれば、これらクラスタ番号に共通するフィールドに“OK”を設定する。ジョブの分散処理が可能でなければ、これらクラスタ番号に共通するフィールドに“NG”を設定する。通常、縦軸のクラスタ番号のプロセッサ群と横軸のクラスタ番号のプロセッサ群とが同じバージョンのデータ転送制御プログラムを実行している場合は“OK”を設定し、同じバージョンのデータ転送制御プログラムを実行できない場合は“NG”を設定する。例えば、全てのクラスタのプロセッサ群が同じバージョンのデータ転送制御プログラムを実行している場合は、図10に示すように全フィールドに“OK”を設定する。一方、クラスタ1のプロセッサ群(211, 213)のみが新しいバージョンのデータ転送制御プログラムの実行を開始しており、クラスタ2～クラスタ4のプロセッサ(212, 214, 221, 223, 222, 224)は古いバージョンのデータ転送制御プログラムの実行を継続している場合は、図11に示すようにクラスタ1に対応するフィールドには“NG”を設定し、クラスタ2～クラスタ4に共通するフィールドには“OK”を設定する。このように、各クラスタのプロセッサ群が実行するデータ転送制御プログラムのバージョンは保守動作時に変化するため、それに合せて分散処理可能クラスタ組合せ情報100の設定値を動的に変更する。

【0036】そして、各プロセッサは、図3の実行クラスタ指定情報600の実行クラスタ指定エリアが“2(クラスタ限定モード)”である場合、図4のI/Oジョブ実行制御情報800の実行可能プロセッサ指定ビットマップエリアの実行可否フィールドのうち、上記分散処理可能クラスタ組合せ情報100で自プロセッサの属するクラスタに対して“OK”が設定されている他クラスタに属するプロセッサに対応した実行可否フィールドにのみ“1(実行可)”を設定する。

【0037】以上のディスク制御装置200'によれば、より細かいクラスタ単位でデータ転送制御プログラムをバージョンアップできる。また、データ転送制御プログラムのバージョンが同一のプロセッサ間でジョブを分散処理できる。

【0038】図12に、プロセッサ群を5つのクラスタにグループ分けした場合の図11相当図を示す。この分散処理可能クラスタ組合せ情報100'は、クラスタ1およびクラスタ2のプロセッサのプログラム変更処理が終了し、クラスタ3のプログラム変更処理を行っている途中であり、クラスタ4およびクラスタ5のプロセッサのプログラム変更処理がまだ開始されていない時点の内容を示している。すなわち、クラスタ1およびクラスタ2のプロセッサは新しいプログラムを実行しているので、クラスタ1およびクラスタ2は分散処理可能な一方

の組になっている。また、クラスタ4およびクラスタ5のプロセッサは古いプログラムを実行しているので、クラスタ4およびクラスタ5は分散処理可能な他方の組になっている。また、クラスタ3は、プログラム変更中なので、他のクラスタとは分散処理可能になっていない。

【0039】-実施例3-

図13は本発明の実施例3のディスク制御装置200"の構成図である。このディスク制御装置200"は、プロセッサ(211, 212, 213, 214)とサービスプロセッサ(215)と共にメモリ(230)とをバス201で接続した構成である。この場合、バス201に接続するプロセッサ(211, 212, 213, 214)を適当なクラスタに分割すれば、上記実施例1や上記実施例2と同様の保守方法によりデータ転送制御プログラムをバージョンアップできる。

【0040】

【発明の効果】本発明のマルチプロセッサシステムの無停止プログラム変更方法によれば、マルチプロセッサシステムの動作を全面停止させることなく、且つ、異なるバージョンのプログラムが干渉することなく、プログラムを変更することが出来る。

【0041】本発明のマルチプロセッサシステムによれば、プログラムの変更のために動作を計画停止させる必要がなくなる。

【0042】本発明のディスク制御装置の無停止プログラム変更方法によれば、ディスク制御装置の動作を全面停止させることなく、且つ、異なるバージョンのデータ転送制御プログラムが干渉することなく、データ転送制御プログラムを変更することが出来る。

【0043】本発明のディスク制御装置によれば、データ転送制御プログラムの変更のために動作を計画停止させる必要がなくなる。

【図面の簡単な説明】

【図1】本発明の実施例1のディスク制御装置の構成図である。

【図2】クラスタ割当て情報の構成図である。

【図3】実行クラスタ指示情報の構成図である。

【図4】I/Oジョブ実行制御情報の構成図である。

【図5】クラスタ動作モード指示情報の構成図である。

【図3】

(図3)

| | |
|----------|---------------------------------|
| 実行クラスタ指示 | 1...クラスタ非限定モード 2...クラスタ限定モード |
|----------|---------------------------------|

600

【図6】無停止保守処理(制御プログラム変更処理)のメインフロー図である。

【図7】クラスタ1の制御プログラム変更処理の詳細フロー図である。

【図8】クラスタ2の制御プログラム変更処理の詳細フロー図である。

【図9】本発明の実施例2のディスク制御装置の構成図である。

【図10】分散処理可能クラスタ組合せ情報の構成図である。

【図11】分散処理可能クラスタ組合せ情報の別の構成図である。

【図12】分散処理可能クラスタ組合せ情報のさらに別の構成図である。

【図13】本発明の実施例3のディスク制御装置の構成図である。

【符号の説明】

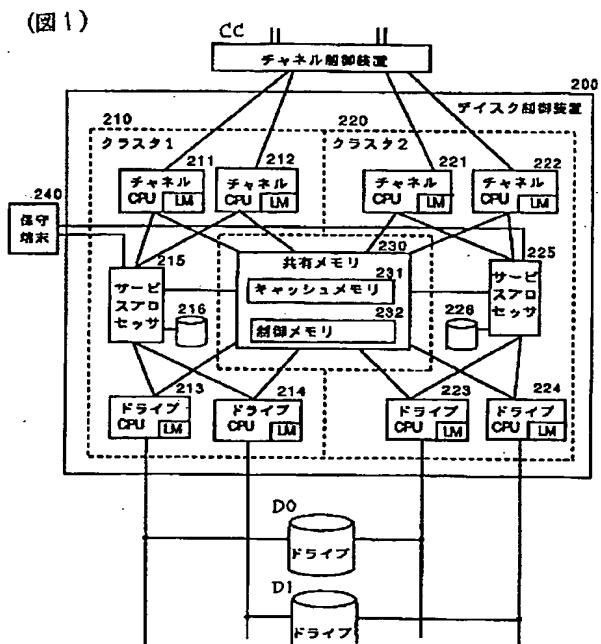
| | |
|--------------------|-----------------|
| 100, 100' | 分散処理可能クラスタ組合せ情報 |
| 200, 200', 200" | ディスク制御装置 |
| 201 | バス |
| 211, 212, 221, 222 | チャネル側プロセッサ |
| 213, 214, 223, 224 | ドライブ側プロセッサ |
| 215, 225 | サービスプロセッサ |
| 230 | 共有メモリ |
| 232 | 制御メモリ |
| 240 | 保守端末 |
| 500 | クラスタ割り当て情報 |
| 600 | 実行クラスタ指示情報 |
| 800 | I/Oジョブ実行制御情報 |
| 700 | クラスタ動作モード指示情報 |
| CC | チャネル制御装置 |
| D0, D1 | ディスクドライブ |

【図10】

(図10) 100 分散処理可能クラスタ組合せ情報

| | クラスタ1 | クラスタ2 | クラスタ3 | クラスタ4 |
|-------|-------|-------|-------|-------|
| クラスタ1 | | OK | OK | OK |
| クラスタ2 | OK | | OK | OK |
| クラスタ3 | OK | OK | | OK |
| クラスタ4 | OK | OK | OK | |

【図1】



【図4】

(図4)

| アクセス スコマ ンド 要求 | コマンド種別 | | | | | | | | | |
|-----------------------------------|--------------------------|------------|-----------------|-----------------|---------|-----|-----|-----|-----|--|
| | ドライブ番号 | | ドライブDO | | | | | | | |
| | シリング番号 | | | | | | | | | |
| | トラック番号 | | | | | | | | | |
| | レコード番号 | | | | | | | | | |
| 実行可能 プロセッサ 指定 ビット マップ | プロセッサ番号 | 211 | 212 | 213 | 214 | 221 | 222 | 223 | 224 | |
| | 実行可否 1...可、 0...不可 | | | | | | | | | |
| I/O要求実行状態 | | 0...実行要求無し | 1...ドライブCPU実行待ち | 2...チャネルCPU実行待ち | 3...実行中 | | | | | |

【図11】

(図11) 100 分散処理可能クラスタ組合せ情報

| | クラスタ1 | クラスタ2 | クラスタ3 | クラスタ4 |
|-------|-------|-------|-------|-------|
| クラスタ1 | | NG | NG | NG |
| クラスタ2 | NG | | OK | OK |
| クラスタ3 | NG | OK | | OK |
| クラスタ4 | NG | OK | OK | |

【図2】

(図2)

クラスタ割り当て情報 500

| プロセッサ番号 | 所属クラスタ番号 | プロセッサ属性 |
|---------|----------|---------|
| 211 | 1 | チャネルCPU |
| 212 | 1 | チャネルCPU |
| 213 | 1 | ドライブCPU |
| 214 | 1 | ドライブCPU |
| 221 | 2 | チャネルCPU |
| 222 | 2 | チャネルCPU |
| 223 | 2 | ドライブCPU |
| 224 | 2 | ドライブCPU |

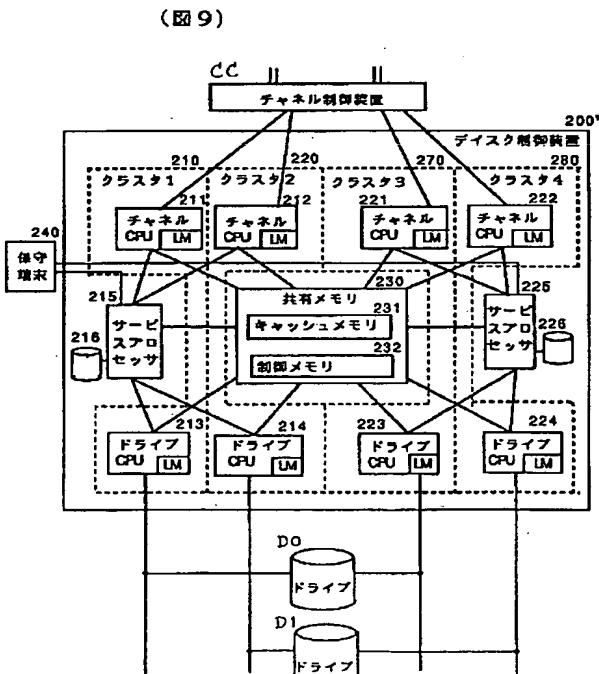
【図5】

(図5)

クラスタ動作モード指示 700

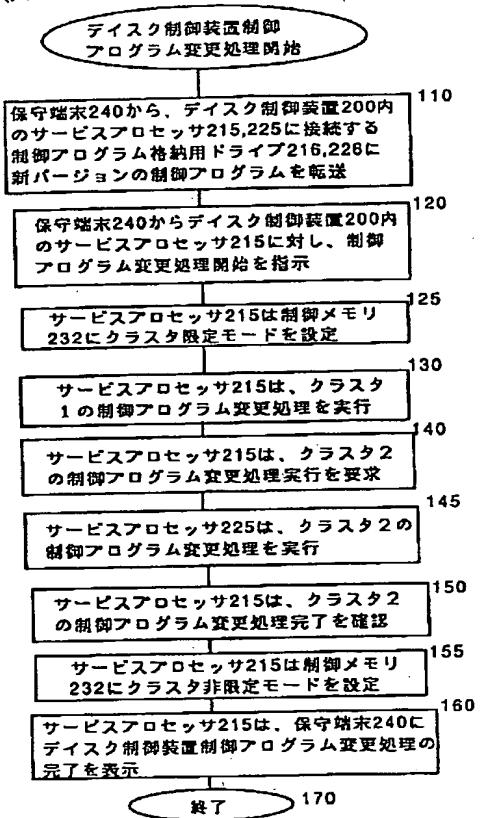
| クラスタ番号 | 動作モード |
|--------|--|
| クラスタ1 | 1... ノーマル動作モード 2... 休止モード 3... 保守動作モード |
| クラスタ2 | |

【図9】



【図6】

(図6)

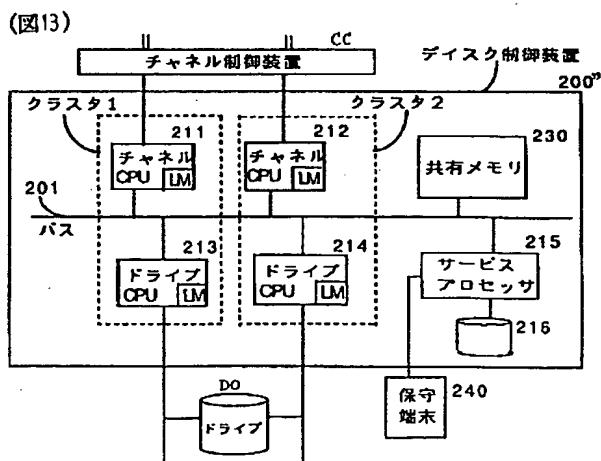


【図12】

(図12) 100' 分散処理可能クラスタ組合せ情報

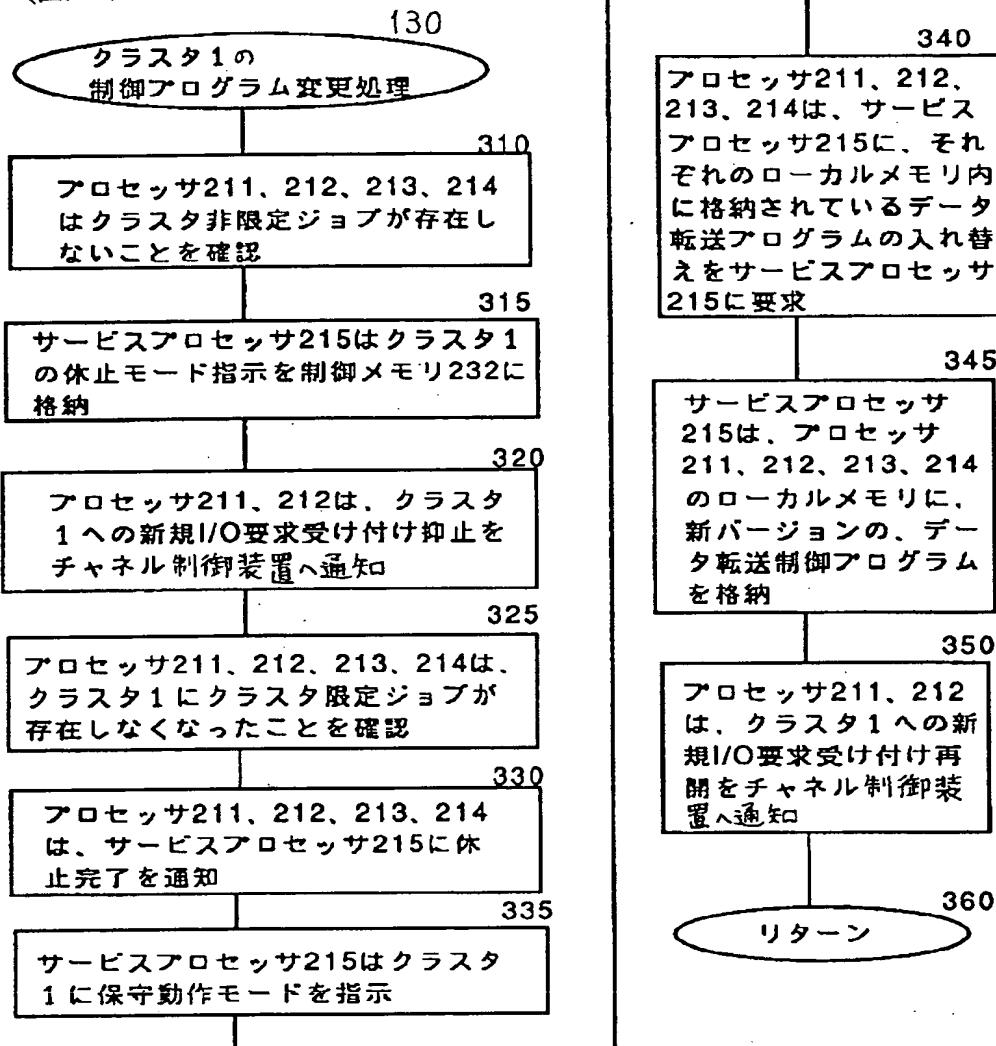
| | クラスタ1 | クラスタ2 | クラスタ3 | クラスタ4 | クラスタ5 |
|-------|-------|-------|-------|-------|-------|
| クラスタ1 | OK | NG | NG | NG | NG |
| クラスタ2 | OK | NG | NG | NG | NG |
| クラスタ3 | NG | NG | NG | NG | NG |
| クラスタ4 | NG | NG | NG | OK | NG |
| クラスタ5 | NG | NG | NG | OK | NG |

【図13】



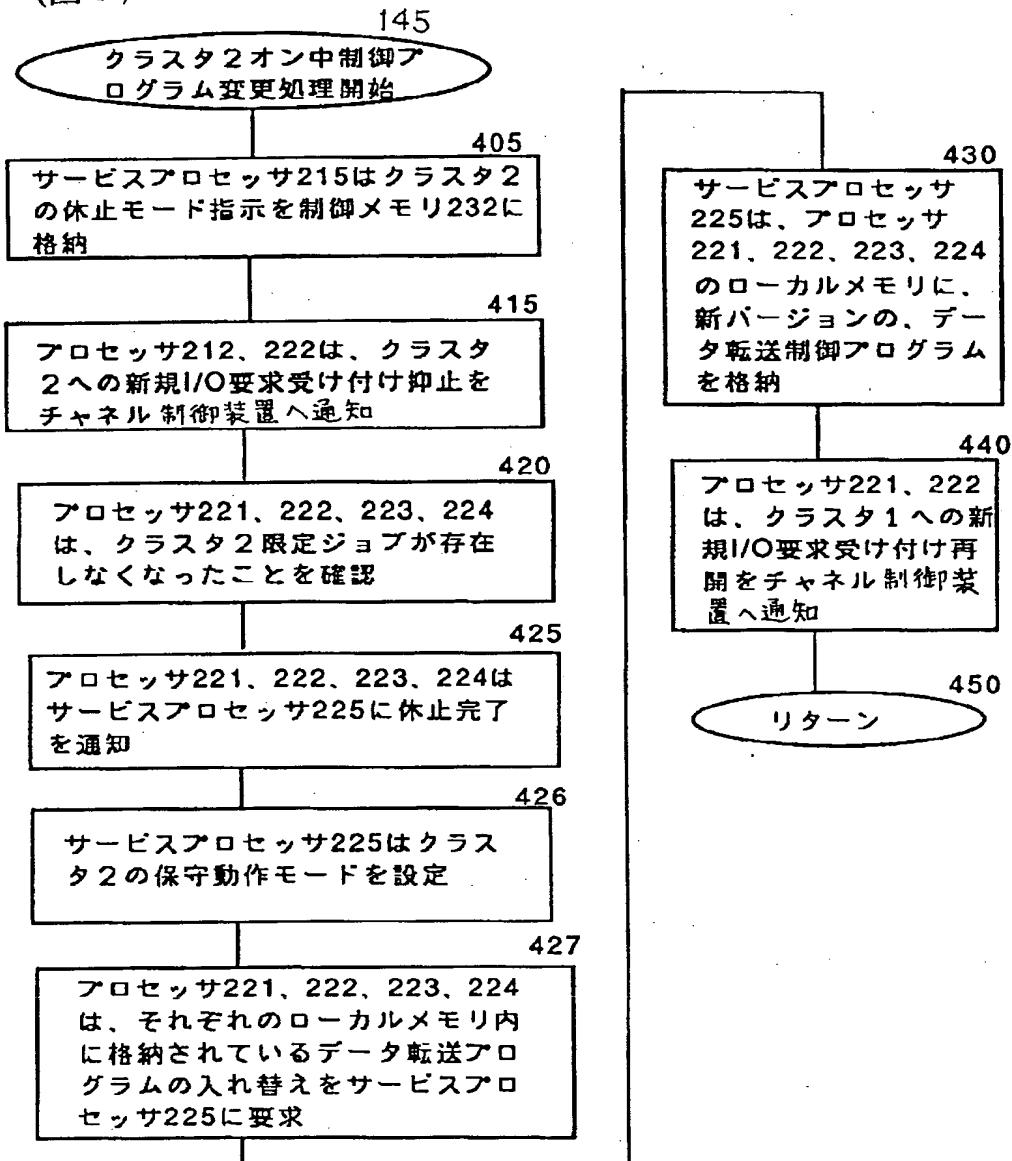
【図7】

(図7)



【図8】

(図8)



【公報種別】特許法第17条の2の規定による補正の掲載

【部門区分】第6部門第3区分

【発行日】平成11年(1999)8月6日

【公開番号】特開平7-306844

【公開日】平成7年(1995)11月21日

【年通号数】公開特許公報7-3069

【出願番号】特願平6-99705

【国際特許分類第6版】

G06F 15/177

9/06 540

【F I】

G06F 15/16 420 S

9/06 540 F

【手続補正書】

【提出日】平成10年8月25日

【手続補正1】

【補正対象書類名】明細書

【補正対象項目名】特許請求の範囲

【補正方法】変更

【補正内容】

【特許請求の範囲】

【請求項1】 複数のプロセッサの各ローカルメモリにそれぞれプログラムを格納し、各プロセッサが自ローカルメモリに格納したプログラムをそれぞれ実行してジョブを処理するマルチプロセッサシステムにおいて、上記複数のプロセッサをクラスタ1からクラスタN(Nは、2以上で全プロセッサ数以下の自然数)までのN個のクラスタにグループ分けし、且つ、自プロセッサが属するクラスタと同一クラスタに属するプロセッサ群の間でのみジョブの分散処理を行なうクラスタ限定モードと全てのプロセッサの間でジョブの分散処理を行なうクラスタ非限定モードとを設け、

通常動作時は、クラスタ非限定モードでジョブを分散処理し、

プログラム変更動作時は、まず、クラスタ非限定モードでのジョブの処理からクラスタ限定モードでのジョブの処理に移行し、移行が完了したら一つのクラスタを選択し、非選択クラスタのプロセッサ群でのジョブの処理および新たなジョブの受け付けを継続しつつ選択クラスタのプロセッサ群での新たなジョブの受け付けを停止し、選択クラスタのプロセッサ群で処理されるジョブが無くなったら選択クラスタのプロセッサ群の各ローカルメモリに格納されているプログラムを新たなプログラムにそれぞれ変更し、選択クラスタの全てのプロセッサのプログラムの変更を完了したら選択クラスタのプロセッサ群での新たなジョブの受け付けおよびジョブの処理を再開し、これを全てのクラスタについて順に繰り返した後、クラスタ限定モードでのジョブの処理をクラスタ非限定

モードでのジョブの処理に戻して通常動作に戻ることを特徴とするマルチプロセッサシステムの無停止プログラム変更方法。

【請求項2】 複数のプロセッサの各ローカルメモリにそれぞれプログラムを格納し、各プロセッサが自ローカルメモリに格納したプログラムをそれぞれ実行してジョブを処理するマルチプロセッサシステムにおいて、上記複数のプロセッサをクラスタ1からクラスタN(Nは、2以上で全プロセッサ数以下の自然数)までのN個のクラスタにグループ分けし、且つ、クラスタ群を第1組と第2組の2個の集合にグループ分けし、且つ、自プロセッサが属するクラスタと同一クラスタに属するプロセッサ群の間でのみジョブの分散処理を行なうクラスタ限定モードと、自プロセッサが属するクラスタと同一クラスタに属するプロセッサ群および自プロセッサが属するクラスタが所属する組と同一組に所属するクラスタに属するプロセッサ群の間でジョブの分散処理を行なうクラスタ非限定モードとを設け、

通常動作時は、全てのクラスタを第1組または第2組の一方に所属させ、クラスタ非限定モードでジョブを分散処理し、

プログラム変更動作時は、まず、一つのクラスタを選択し、非選択クラスタにおけるクラスタ非限定モードでのジョブの分散処理を継続しつつ選択クラスタにおけるクラスタ非限定モードでのジョブの処理をクラスタ限定モードでのジョブの処理へと移行し、移行が完了したら選択クラスタのプロセッサ群での新たなジョブの受け付けを停止し、選択クラスタのプロセッサ群で処理されるジョブが無くなったら選択クラスタのプロセッサ群の各ローカルメモリに格納されているプログラムを新たなプログラムにそれぞれ変更し、選択クラスタの全てのプロセッサのプログラムの変更を完了したら選択クラスタのプロセッサ群での新たなジョブの受け付けおよびジョブの処理を再開すると共に、選択クラスタの所属を第1組ま

たは第2組の一方から他方に変更した上で選択クラスタにおけるクラスタ限定モードでのジョブの処理をクラスタ非限定モードでのジョブの処理に戻し、これを全てのクラスタについて順に繰り返して通常動作に戻ることを特徴とするマルチプロセッサシステムの無停止プログラム変更方法。

【請求項3】 複数のプロセッサの各ローカルメモリにそれぞれプログラムを格納し、各プロセッサが自ローカルメモリに格納したプログラムをそれぞれ実行してジョブを処理するマルチプロセッサシステムにおいて、上記複数のプロセッサをクラスタ1からクラスタN（Nは、2以上で全プロセッサ数以下の自然数）までのN個のクラスタにグループ分けするクラスタ割り当て手段と、

自プロセッサが属するクラスタと同一クラスタに属するプロセッサ群の間でのみジョブの分散処理を行うクラスタ限定モードか、全てのプロセッサの間でジョブの分散処理を行なうクラスタ非限定モードか、のいずれかで分散処理を行なうよう制御する分散処理モード制御手段と、通常動作時は、前記クラスタ非限定モードによりジョブの分散処理を行なわせ、プログラム変更動作時は、まず、クラスタ非限定モードでのジョブの処理からクラスタ限定モードでのジョブの処理に移行させ、移行が完了したら一つのクラスタを選択し、非選択クラスタのプロセッサ群でのジョブの処理および新たなジョブの受け付けを継続させつつ選択クラスタのプロセッサ群での新たなジョブの受け付けを停止させ、選択クラスタのプロセッサ群で処理されるジョブが無くなったら選択クラスタのプロセッサ群の各ローカルメモリに格納されているプログラムを新たなプログラムにそれぞれ変更し、選択クラスタの全てのプロセッサのプログラムの変更を完了したら選択クラスタのプロセッサ群での新たなジョブの受け付けおよびジョブの処理を再開させ、これを全てのクラスタについて順に繰り返した後、クラスタ限定モードでのジョブの処理をクラスタ非限定モードでのジョブの処理に移行させ通常動作に戻らせるプログラム変更処理手段とを具備したことを特徴とするマルチプロセッサシステム。

【請求項4】 複数のプロセッサの各ローカルメモリにそれぞれプログラムを格納し、各プロセッサが自ローカルメモリに格納したプログラムをそれぞれ実行してジョブを処理するマルチプロセッサシステムにおいて、上記複数のプロセッサをクラスタ1からクラスタN（Nは、2以上で全プロセッサ数以下の自然数）までのN個のクラスタにグループ分けするクラスタ割り当て手段と、

クラスタ群を第1組と第2組の2個の集合にグループ分けする分散処理可能クラスタ組合せ手段と、自プロセッサが属するクラスタと同一クラスタに属するプロセッサ群の間でのみジョブの分散処理を行なうクラ

スタ限定モードか、自プロセッサが属するクラスタと同一クラスタに属するプロセッサ群および自プロセッサが属するクラスタが所属する組と同一組に所属するクラスタに属するプロセッサ群の間でジョブの分散処理を行なうクラスタ非限定モードか、のいずれかで分散処理を行うよう制御する分散処理モード制御手段と、

通常動作時は、全てのクラスタを第1組または第2組の一方に所属させ、クラスタ非限定モードでジョブを分散処理させ、プログラム変更動作時は、まず、一つのクラスタを選択し、非選択クラスタにおけるクラスタ非限定モードでのジョブの分散処理を継続させつつ選択クラスタにおけるクラスタ非限定モードでのジョブの処理をクラスタ限定モードでのジョブの処理へと移行させ、移行が完了したら選択クラスタのプロセッサ群での新たなジョブの受け付けを停止させ、選択クラスタのプロセッサ群で処理されるジョブが無くなったら選択クラスタのプロセッサ群の各ローカルメモリに格納されているプログラムを新たなプログラムにそれぞれ変更し、選択クラスタの全てのプロセッサのプログラムの変更を完了したら選択クラスタのプロセッサ群での新たなジョブの受け付けおよびジョブの処理を再開させると共に、選択クラスタの所属を第1組または第2組の一方から他方に変更させた上で選択クラスタにおけるクラスタ限定モードでのジョブの処理をクラスタ非限定モードでのジョブの処理に移行させ、これを全てのクラスタについて順に繰り返して通常動作に戻らせるプログラム変更処理手段とを具備したことを特徴とするマルチプロセッサシステム。

【請求項5】 複数のプロセッサの各ローカルメモリにそれぞれプログラムを格納し、各プロセッサが自ローカルメモリに格納したプログラムをそれぞれ実行してジョブを処理するマルチプロセッサシステムにおいて、上記複数のプロセッサを2以上のクラスタにグループ分けするために各プロセッサがいずれのクラスタに属しているかを示すクラスタ割り当て情報を記憶するクラスタ割り当て情報記憶手段と、

ローカルメモリに格納したプログラムに従ってプロセッサがジョブの処理を行うノーマル動作モードか、クラスタを開塞するためにプロセッサがジョブの処理を休止する休止モードか、ローカルメモリに格納されたプログラムを新たなプログラムに変更する保守動作モードか、のいずれの動作モードにするかをクラスタ単位で示すクラスタ動作モード指示情報を記憶する動作モード指示情報記憶手段と、

自プロセッサが属するクラスタと同じクラスタに属するプロセッサとのみジョブの分散処理を行なうクラスタ限定モードか、クラスタに拘り無く任意のプロセッサ間でジョブの分散処理を行なうクラスタ非限定モードか、のいずれの分散処理モードでジョブを処理するかを示す実行クラスタ指示情報を記憶する実行クラスタ指示情報記憶手段と、

プログラムの変更開始を指示するプログラム変更開始指示手段と、

プログラムの変更開始が指示されたとき、分散処理モードをクラスタ限定モードに設定し、任意のクラスタを一つ選び、そのクラスタの動作モードを休止モードに設定し、当該クラスタに属するプロセッサが全て休止状態になった時点で当該クラスタの動作モードを保守動作モードに設定し、当該クラスタに属する全てのプロセッサのローカルメモリに格納されているプログラムを新たなプログラムに変更し、これを全クラスタについて繰り返し、その完了時点で全てのクラスタの動作モードをノーマル動作モードに設定し、分散処理モードをクラスタ非限定モードに設定するプログラム変更処理手段と、を具備したことを特徴とするマルチプロセッサシステム。

【請求項6】 複数のプロセッサの各ローカルメモリにそれぞれプログラムを格納し、各プロセッサが自ローカルメモリに格納したプログラムをそれぞれ実行してジョブを処理するマルチプロセッサシステムにおいて、上記複数のプロセッサを2以上のクラスタにグループ分けするために各プロセッサがいずれのクラスタに属しているかを示すクラスタ割当て情報を記憶するクラスタ割当て情報記憶手段と、

ローカルメモリに格納したプログラムに従ってプロセッサがジョブの処理を行うノーマル動作モードか、クラスタを閉塞するためにプロセッサがジョブの処理を休止する休止モードか、ローカルメモリに格納されたプログラムを新たなプログラムに変更する保守動作モードか、のいずれの動作モードにするかをクラスタ単位で示すクラスタ動作モード指示情報を記憶する動作モード指示情報記憶手段と、

自プロセッサが属するクラスタと同じクラスタに属するプロセッサとのみジョブの分散処理を行なうクラスタ限定モードか、自プロセッサが属するクラスタと同じクラスタに属するプロセッサおよび自プロセッサとの間でジョブの分散処理が可能な他クラスタのプロセッサでジョブの分散処理を行なうクラスタ非限定モードか、のいずれの分散処理モードでジョブを処理するかを示す実行クラスタ指示情報を記憶する実行クラスタ指示情報記憶手段と、

プログラムの変更開始を指示するプログラム変更開始指示手段と、

プログラムの変更開始が指示されたとき、任意のクラスタを一つ選び、そのクラスタの分散処理モードをクラスタ限定モードに設定し、当該クラスタの動作モードを休止モードに設定し、当該クラスタに属するプロセッサが全て休止状態になった時点で当該クラスタの動作モードを保守動作モードに設定し、当該クラスタに属する全てのプロセッサのローカルメモリに格納されているプログラムを新たなプログラムに変更し、当該クラスタの分散処理モードをクラスタ非限定モードに設定し、これを全

クラスタについて繰り返し、その完了時点で全てのクラスタの動作モードをノーマル動作モードに設定するプログラム変更処理手段と、を具備したことを特徴とするマルチプロセッサシステム。

【請求項7】 複数のプロセッサの各ローカルメモリにそれぞれデータ転送制御プログラムを格納し、各プロセッサが自ローカルメモリに格納したデータ転送制御プログラムをそれぞれ実行してディスクドライブに対するデータ転送ジョブを処理するディスク制御装置において、上記複数のプロセッサをクラスタ1からクラスタN(Nは、2以上で全プロセッサ数以下の自然数)までのN個のクラスタにグループ分けし、且つ、自プロセッサが属するクラスタと同一クラスタに属するプロセッサ群の間でのみデータ転送ジョブの分散処理を行なうクラスタ限定モードと全てのプロセッサの間でデータ転送ジョブの分散処理を行なうクラスタ非限定モードとを設け、通常動作時は、クラスタ非限定モードでデータ転送ジョブを分散処理し、

データ転送制御プログラム変更動作時は、まず、クラスタ非限定モードでのデータ転送ジョブの処理からクラスタ限定モードでのデータ転送ジョブの処理に移行し、移行が完了したら一つのクラスタを選択し、非選択クラスタのプロセッサ群でのデータ転送ジョブの処理および新たなデータ転送ジョブの受け付けを継続しつつ選択クラスタのプロセッサ群での新たなデータ転送ジョブの受け付けを停止し、選択クラスタのプロセッサ群で処理されるデータ転送ジョブが無くなったら選択クラスタのプロセッサ群の各ローカルメモリに格納されているデータ転送制御プログラムを新たなデータ転送制御プログラムにそれぞれ変更し、選択クラスタの全てのプロセッサのデータ転送制御プログラムの変更を完了したら選択クラスタのプロセッサ群での新たなデータ転送ジョブの受け付けおよびデータ転送ジョブの処理を再開し、これを全てのクラスタについて順に繰り返した後、クラスタ限定モードでのデータ転送ジョブの処理をクラスタ非限定モードでのデータ転送ジョブの処理に戻して通常動作に戻ることを特徴とするディスク制御装置の無停止プログラム変更方法。

【請求項8】 複数のプロセッサの各ローカルメモリにそれぞれデータ転送制御プログラムを格納し、各プロセッサが自ローカルメモリに格納したデータ転送制御プログラムをそれぞれ実行してデータ転送ジョブを処理するディスク制御装置において、

上記複数のプロセッサをクラスタ1からクラスタN(Nは、2以上で全プロセッサ数以下の自然数)までのN個のクラスタにグループ分けし、且つ、クラスタ群を第1組と第2組の2個の集合にグループ分けし、且つ、自プロセッサが属するクラスタと同一クラスタに属するプロセッサ群の間でのみデータ転送ジョブの分散処理を行なうクラスタ限定モードと、自プロセッサが属するクラス

タと同一クラスタに属するプロセッサ群および自プロセッサが属するクラスタが所属する組と同一組に所属するクラスタに属するプロセッサ群の間でデータ転送ジョブの分散処理を行なうクラスタ非限定モードとを設け、通常動作時は、全てのクラスタを第1組または第2組の一方に所属させ、クラスタ非限定モードでデータ転送ジョブを分散処理し、

データ転送制御プログラム変更動作時は、まず、一つのクラスタを選択し、非選択クラスタにおけるクラスタ非限定モードでのデータ転送ジョブの分散処理を継続しつつ選択クラスタにおけるクラスタ非限定モードでのデータ転送ジョブの処理をクラスタ限定モードでのデータ転送ジョブの処理へと移行し、移行が完了したら選択クラスタのプロセッサ群での新たなデータ転送ジョブの受け付けを停止し、選択クラスタのプロセッサ群で処理されるデータ転送ジョブが無くなったら選択クラスタのプロセッサ群の各ローカルメモリに格納されているデータ転送制御プログラムを新たなデータ転送制御プログラムにそれぞれ変更し、選択クラスタの全てのプロセッサのデータ転送制御プログラムの変更を完了したら選択クラスタのプロセッサ群での新たなデータ転送ジョブの受け付けおよびデータ転送ジョブの処理を再開すると共に、選択クラスタの所属を第1組または第2組の一方から他方に変更した上で選択クラスタにおけるクラスタ限定モードでのデータ転送ジョブの処理をクラスタ非限定モードでのデータ転送ジョブの処理に戻し、これを全てのクラスタについて順に繰り返して通常動作に戻ることを特徴とするディスク制御装置の無停止プログラム変更方法。

【請求項9】 複数のプロセッサの各ローカルメモリにそれぞれデータ転送制御プログラムを格納し、各プロセッサが自ローカルメモリに格納したデータ転送制御プログラムをそれぞれ実行してデータ転送ジョブを処理するディスク制御装置において、

上記複数のプロセッサをクラスタ1からクラスタN（Nは、2以上で全プロセッサ数以下の自然数）までのN個のクラスタにグループ分けするクラスタ割り当て手段と、

自プロセッサが属するクラスタと同一クラスタに属するプロセッサ群の間でのみデータ転送ジョブの分散処理を行うクラスタ限定モードか、全てのプロセッサの間でデータ転送ジョブの分散処理を行なうクラスタ非限定モードか、のいずれかで分散処理を行うよう制御する分散処理モード制御手段と、

通常動作時は、前記クラスタ非限定モードによりデータ転送ジョブの分散処理を行なわせ、データ転送制御プログラム変更動作時は、まず、クラスタ非限定モードでのデータ転送ジョブの処理からクラスタ限定モードでのデータ転送ジョブの処理に移行させ、移行が完了したら一つのクラスタを選択し、非選択クラスタのプロセッサ群でのデータ転送ジョブの処理および新たなデータ転送

ジョブの受け付けを継続させつつ選択クラスタのプロセッサ群での新たなデータ転送ジョブの受け付けを停止させ、選択クラスタのプロセッサ群で処理されるデータ転送ジョブが無くなったら選択クラスタのプロセッサ群の各ローカルメモリに格納されているデータ転送制御プログラムを新たなデータ転送制御プログラムにそれぞれ変更し、選択クラスタの全てのプロセッサのデータ転送制御プログラムの変更を完了したら選択クラスタのプロセッサ群での新たなデータ転送ジョブの受け付けおよびデータ転送ジョブの処理を再開させ、これを全てのクラスタについて順に繰り返した後、クラスタ限定モードでのデータ転送ジョブの処理をクラスタ非限定モードでのデータ転送ジョブの処理に移行させ通常動作に戻らせるデータ転送制御プログラム変更処理手段とを具備したことを特徴とするディスク制御装置。

【請求項10】 複数のプロセッサの各ローカルメモリにそれぞれデータ転送制御プログラムを格納し、各プロセッサが自ローカルメモリに格納したデータ転送制御プログラムをそれぞれ実行してデータ転送ジョブを処理するディスク制御装置において、

上記複数のプロセッサをクラスタ1からクラスタN（Nは、2以上で全プロセッサ数以下の自然数）までのN個のクラスタにグループ分けするクラスタ割り当て手段と、

クラスタ群を第1組と第2組の2個の集合にグループ分けする分散処理可能クラスタ組合せ手段と、自プロセッサが属するクラスタと同一クラスタに属するプロセッサ群とのみデータ転送ジョブの分散処理を行なうクラスタ限定モードか、自プロセッサが属するクラスタと同一クラスタに属するプロセッサ群および自プロセッサが属するクラスタが所属する組と同一組に所属するクラスタに属するプロセッサ群の間でデータ転送ジョブの分散処理を行なうクラスタ非限定モードか、のいずれかで分散処理を行うよう制御する分散処理モード制御手段と、

通常動作時は、全てのクラスタを第1組または第2組の一方に所属させ、クラスタ非限定モードでデータ転送ジョブを分散処理させ、データ転送制御プログラム変更動作時は、まず、一つのクラスタを選択し、非選択クラスタにおけるクラスタ非限定モードでのデータ転送ジョブの分散処理を継続させつつ選択クラスタにおけるクラスタ非限定モードでのデータ転送ジョブの処理をクラスタ限定モードでのデータ転送ジョブの処理へと移行させ、移行が完了したら選択クラスタのプロセッサ群での新たなデータ転送ジョブの受け付けを停止させ、選択クラスタのプロセッサ群で処理されるデータ転送ジョブが無くなったら選択クラスタのプロセッサ群の各ローカルメモリに格納されているデータ転送制御プログラムを新たなデータ転送制御プログラムにそれぞれ変更し、選択クラスタの全てのプロセッサのデータ転送制御プログラムの

変更を完了したら選択クラスタのプロセッサ群での新たなデータ転送ジョブの受け付けおよびデータ転送ジョブの処理を再開させると共に、選択クラスタの所属を第1組または第2組の一方から他方に変更させた上で選択クラスタにおけるクラスタ限定モードでのデータ転送ジョブの処理をクラスタ非限定モードでのデータ転送ジョブの処理に移行させ、これを全てのクラスタについて順に繰り返して通常動作に戻らせるデータ転送制御プログラム変更処理手段とを具備したことを特徴とするディスク制御装置。

【請求項11】 複数のプロセッサの各ローカルメモリにそれぞれデータ転送制御プログラムを格納し、各プロセッサが自ローカルメモリに格納したデータ転送制御プログラムをそれぞれ実行してデータ転送ジョブを処理するディスク制御装置において、

上記複数のプロセッサを2以上のクラスタにグループ分けするために各プロセッサがいずれのクラスタに属しているかを示すクラスタ割当情報を記憶するクラスタ割当情報記憶手段と、

ローカルメモリに格納したデータ転送制御プログラムに従ってプロセッサがデータ転送ジョブの処理を行うノーマル動作モードか、クラスタを閉塞するためにプロセッサがデータ転送ジョブの処理を休止する休止モードか、ローカルメモリに格納されたデータ転送制御プログラムを新たなデータ転送制御プログラムに変更する保守動作モードか、のいずれの動作モードにするかをクラスタ単位で示すクラスタ動作モード指示情報を記憶する動作モード指示情報記憶手段と、

自プロセッサが属するクラスタと同じクラスタに属するプロセッサとのみデータ転送ジョブの分散処理を行なうクラスタ限定モードか、クラスタに拘り無く任意のプロセッサ間でデータ転送ジョブの分散処理を行なうクラスタ非限定モードか、のいずれの分散処理モードでデータ転送ジョブを処理するかを示す実行クラスタ指示情報を記憶する実行クラスタ指示情報記憶手段と、

データ転送制御プログラムの変更開始を指示するデータ転送制御プログラム変更開始指示手段と、

データ転送制御プログラムの変更開始が指示されたとき、分散処理モードをクラスタ限定モードに設定し、任意のクラスタを一つ選び、そのクラスタの動作モードを休止モードに設定し、当該クラスタに属するプロセッサが全て休止状態になった時点で当該クラスタの動作モードを保守動作モードに設定し、当該クラスタに属する全てのプロセッサのローカルメモリに格納されているデータ転送制御プログラムを新たなデータ転送制御プログラムに変更し、これを全クラスタについて繰り返し、その完了時点で全てのクラスタの動作モードをノーマル動作モードに設定し、分散処理モードをクラスタ非限定モードに設定するデータ転送制御プログラム変更処理手段と、を具備したことを特徴とするディスク制御装置。

【請求項12】 複数のプロセッサの各ローカルメモリにそれぞれデータ転送制御プログラムを格納し、各プロセッサが自ローカルメモリに格納したデータ転送制御プログラムをそれぞれ実行してデータ転送ジョブを処理するディスク制御装置において、

上記複数のプロセッサを2以上のクラスタにグループ分けするために各プロセッサがいずれのクラスタに属しているかを示すクラスタ割当情報を記憶するクラスタ割当情報記憶手段と、

ローカルメモリに格納したデータ転送制御プログラムに従ってプロセッサがデータ転送ジョブの処理を行なうノーマル動作モードか、クラスタを閉塞するためにプロセッサがデータ転送ジョブの処理を休止する休止モードか、ローカルメモリに格納されたデータ転送制御プログラムを新たなデータ転送制御プログラムに変更する保守動作モードか、のいずれの動作モードにするかをクラスタ単位で示すクラスタ動作モード指示情報を記憶する動作モード指示情報記憶手段と、

自プロセッサが属するクラスタと同じクラスタに属するプロセッサとのみデータ転送ジョブの分散処理を行なうクラスタ限定モードか、自プロセッサが属するクラスタと同じクラスタに属するプロセッサおよび自プロセッサとの間でデータ転送ジョブの分散処理が可能な他クラスタのプロセッサでデータ転送ジョブの分散処理を行なうクラスタ非限定モードか、のいずれの分散処理モードでデータ転送ジョブを処理するかを示す実行クラスタ指示情報を記憶する実行クラスタ指示情報記憶手段と、

データ転送制御プログラムの変更開始を指示するデータ転送制御プログラム変更開始指示手段と、

データ転送制御プログラムの変更開始が指示されたとき、任意のクラスタを一つ選び、そのクラスタの分散処理モードをクラスタ限定モードに設定し、当該クラスタの動作モードを休止モードに設定し、当該クラスタに属するプロセッサが全て休止状態になった時点で当該クラスタの動作モードを保守動作モードに設定し、当該クラスタに属する全てのプロセッサのローカルメモリに格納されているデータ転送制御プログラムを新たなデータ転送制御プログラムに変更し、当該クラスタの分散処理モードをクラスタ非限定モードに設定し、これを全クラスタについて繰り返し、その完了時点で全てのクラスタの動作モードをノーマル動作モードに設定するデータ転送制御プログラム変更処理手段と、を具備したことを特徴とするディスク制御装置。

【請求項13】 ディスク制御装置であって、
ホストプロセッサからの入出力要求を受け取るチャネルプロセッサと、

前記チャネルプロセッサから前記入出力要求に対応するデータアクセス要求を受け取り、前記データアクセス要求で指定されたデータをドライブとの間で転送するドライブプロセッサと、

前記チャネルプロセッサと前記ドライブプロセッサとが接続される共通バスと、

前記チャネルプロセッサと前記ドライブプロセッサとに接続され、前記チャネルプロセッサと前記ドライブプロセッサとにプログラムを供給するサービスプロセッサとから構成され、

前記チャネルプロセッサと前記ドライブプロセッサは、前記入出力要求を、新しいプログラムをローディング中は旧プログラムを使って処理し、

前記チャネルプロセッサと前記ドライブプロセッサは、前記入出力要求を、前記新しいプログラムが前記チャネルプロセッサと前記ドライブプロセッサの両方において使用可能となった後は、前記新しいプログラムを使って処理する、ことを特徴とするディスク制御装置。

【請求項14】 複数の各プロセッサに新しいプログラムをローディング中は旧プログラムを使用して動作可能なディスク制御装置であって、

ホストプロセッサからの入出力要求を受け取るチャネルプロセッサと、

前記チャネルプロセッサから前記入出力要求に対応するデータアクセス要求を受け取り、前記データアクセス要求で指定されたデータをドライブとの間で転送するドライブプロセッサとから構成され、

前記チャネルプロセッサと前記ドライブプロセッサは、前記入出力要求を、新しいプログラムをローディング中は旧プログラムを使って処理し、

前記チャネルプロセッサと前記ドライブプロセッサは、前記入出力要求を、前記新しいプログラムが前記チャネルプロセッサと前記ドライブプロセッサの両方において使用可能となった後は、前記新しいプログラムを使って処理する、ことを特徴とするディスク制御装置。

【手続補正2】

【補正対象書類名】明細書

【補正対象項目名】0009

【補正方法】変更

【補正内容】

【0009】第6の観点では、本発明は、上記第4の観点による無停止プログラム変更方法または上記第5の観点による無停止プログラム変更方法を実施するディスク制御装置を提供する。第7の観点では、本発明は、ディスク制御装置であって、ホストプロセッサからの入出力要求を受け取るチャネルプロセッサと、前記チャネルプロセッサから前記入出力要求に対応するデータアクセス要求を受け取り、前記データアクセス要求で指定されたデータをドライブとの間で転送するドライブプロセッサと、前記チャネルプロセッサと前記ドライブプロセッサ

とが接続される共通バスと、前記チャネルプロセッサと前記ドライブプロセッサとに接続され、前記チャネルプロセッサと前記ドライブプロセッサとにプログラムを供給するサービスプロセッサとから構成され、前記チャネルプロセッサと前記ドライブプロセッサは、前記入出力要求を、新しいプログラムをローディング中は旧プログラムを使って処理し、前記チャネルプロセッサと前記ドライブプロセッサは、前記入出力要求を、前記新しいプログラムが前記チャネルプロセッサと前記ドライブプロセッサの両方において使用可能となった後は、前記新しいプログラムを使って処理することを特徴とするディスク制御装置を提供する。第8の観点では、本発明は、複数の各プロセッサに新しいプログラムをローディング中は旧プログラムを使用して動作可能なディスク制御装置であって、ホストプロセッサからの入出力要求を受け取るチャネルプロセッサと、前記チャネルプロセッサから前記入出力要求に対応するデータアクセス要求を受け取り、前記データアクセス要求で指定されたデータをドライブとの間で転送するドライブプロセッサとから構成され、前記チャネルプロセッサと前記ドライブプロセッサは、前記入出力要求を、新しいプログラムをローディング中は旧プログラムを使って処理し、前記チャネルプロセッサと前記ドライブプロセッサは、前記入出力要求を、前記新しいプログラムが前記チャネルプロセッサと前記ドライブプロセッサの両方において使用可能となった後は、前記新しいプログラムを使って処理することを特徴とするディスク制御装置を提供する。

【手続補正3】

【補正対象書類名】明細書

【補正対象項目名】0015

【補正方法】変更

【補正内容】

【0015】上記第6の観点によるディスク制御装置では、上記無停止データ転送制御プログラム変更方法を好適に実施可能であるから、データ転送制御プログラムを変更するときに動作を全面停止させずに済む。上記第7の観点および上記第8の観点によるディスク制御装置では、新しいプログラムがチャネルプロセッサとドライブプロセッサの両方において使用可能となる前は旧プログラムを使用して入出力要求を処理し、前記新しいプログラムが前記チャネルプロセッサと前記ドライブプロセッサの両方において使用可能となった後は、前記新しいプログラムを使用して入出力要求を処理するから、プログラムを変更するときに入出力要求の処理を停止させずに済む。